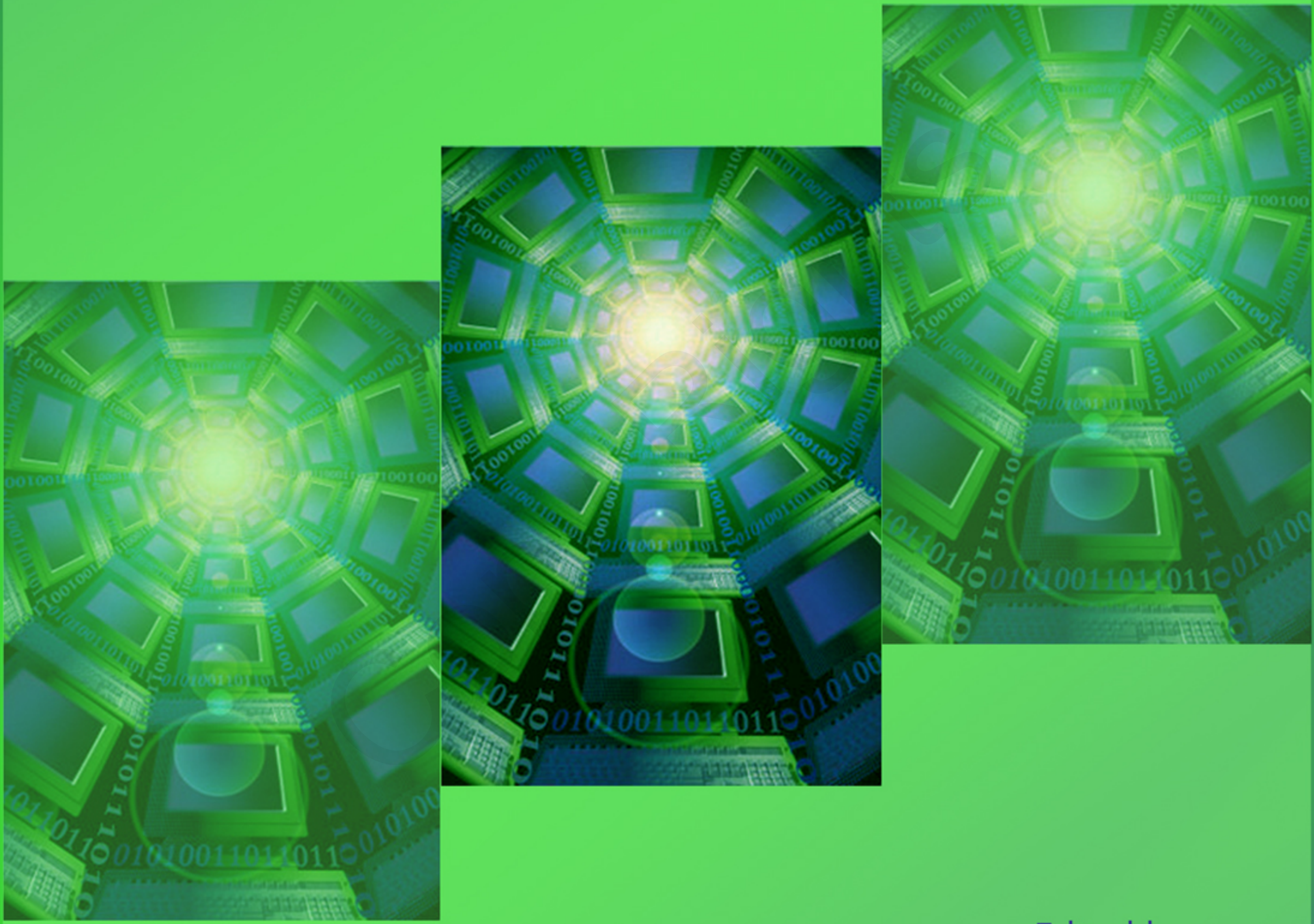
Proceedings of the IADIS International Conference

APPLIED COMPUTING 2010

Timisoara, Romania 14-16 October



Edited by
Hans Weghorn
Pedro Isaías
Radu Vasiu

iadis
international association for development of the information society

Co-organised by:

IADIS INTERNATIONAL CONFERENCE

APPLIED COMPUTING 2010

Copy Rights

Copy Rights

PROCEEDINGS OF THE
IADIS INTERNATIONAL CONFERENCE
APPLIED COMPUTING 2010

TIMISOARA, ROMANIA

14-16 October 2010

Copy Rights

Organised by

IADIS

International Association for Development of the Information Society

Co-Organised by

Copyright 2010
IADIS Press
All rights reserved

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Permission for use must always be obtained from IADIS Press. Please contact secretariat@iadis.org

Edited by Hans Weghorn, Pedro Isaías and Radu Vasiu

Associate Editors: Luís Rodrigues and Patrícia Barbosa

ISBN: 978-972-8939-30-4

Copy Rights

TABLE OF CONTENTS

FULL PAPERS

Copy Rights

Copy Rights

SHORT PAPERS

Copy Rights

REFLECTION PAPERS

POSTERS

Copy Rights

FOREWORD

These proceedings contain the papers of the IADIS Applied Computing 2010, which was organised by the International Association for Development of the Information Society and co-organised by "Politehnica" University of Timisoara, Romania, 14-16 October.

The IADIS Applied Computing 2010 conference aims to address the main issues of concern within the applied computing area and related fields. This conference covers essentially technical aspects. The applied computing field is divided into more detailed areas.

The following thirty-six areas have been object of paper and poster submissions:

Agent Systems and Applications, Algorithms, Applied Information Systems, Bioinformatics, Case Studies and Applications, Communications, Data Mining, Database Systems, E-Commerce Theory and Practice, Embedded Systems, Evaluation and Assessment, Global Tendencies, Grid Computing, Information Retrieval, Intelligent Systems, Mobile Networks and Systems, Multimedia, Networking, Object Orientation, Parallel and Distributed Systems, Payment Systems, Programming Languages, Protocols and Standards, Security, Semantic Web, Software Engineering, Storage Issues, Technologies for E-Learning, Wireless Applications, WWW Applications, WWW Technologies, Ubiquitous Computing, Usability Issues, Virtual Reality, Visualization, XML and other Extensible Languages

The IADIS Applied Computing 2010 Conference had 130 submissions from 26 countries. Each submission has been anonymously reviewed by an average of five independent reviewers, to ensure the final high standard of the accepted submissions. Out of the papers submitted, 25 got blind referee ratings that published them as full papers, which means that the acceptance rate was below 20%. Some other submissions were published as short papers, reflection papers, and posters. Authors of the best published papers in the Applied Computing 2010 proceedings will be invited to publish extended versions of their papers in the IADIS International Journal on Computer Science and Information Systems (IJCSIS) (ISSN 1646-3692) and other selected journals.

The conference, besides the presentation of full papers, short papers, reflection papers, and posters also includes two keynote presentations from internationally distinguished researchers: we wish to thank Professor Vasile Baltac, ATIC - IT&C Association of Romania, Vice-Chairman of World Information Technology and Software Alliance (WITSA) and President of the Council of European Professional and Informatics Societies (CEPIS), Romania, and Dr. Alina Andreica, Associate Professor, Head of IT Department, Babes-Bolyai University, Cluj-Napoca, Romania.

Copy Rights

As we all know, a conference requires the effort of many individuals. We would like to thank all members of the Program Committee for they hard work in reviewing and selecting the papers that appear in the book of the proceedings. Special thanks also to the auxiliary reviewers that contributed to the reviewing process.

Last but not the least, we hope that everybody will have a good time in Timisoara, and we invite all participants for the next edition of the IADIS International Conference Applied Computing 2011.

Hans Weghorn, BW Cooperative State University Stuttgart, Germany
Program Chair

Pedro Isaías, Universidade Aberta (Portuguese Open University), Portugal
Radu Vasiu, "Politehnica" University of Timisoara, Romania
Conference Co-Chairs

Timisoara, Romania
14 October 2010

Copy Rights

PROGRAM COMMITTEE

PROGRAM CHAIR

Hans Weghorn, BW Cooperative State University Stuttgart, Germany

CONFERENCE CO-CHAIRS

Pedro Isaías, Universidade Aberta (Portuguese Open University), Portugal
Radu Vasiu, "Politehnica" University of Timisoara, Romania

COMMITTEE MEMBERS

Adam Wong, Deakin University, Australia
Aijuan Dong, Hood College, USA
Alan Barton, IIT -- National Research Council, Canada
Alberto Ros, Universidad de Murcia, Spain
Ali Masoudi-Nejad, Laboratory of Systems Biology and Bioinformatics, Iran
Ali Shiri, University Of Alberta, Canada
Ana Carolina Lorena, Universidade Federal do ABC, Brazil
Anastasios Doulamis, Technical University Of Crete, Greece
Andreas Andreou, University Of Cyprus, Cyprus
Andres Muñoz, Universidad De Murcia, Spain
Anne-muriel Arigon, Université Montpellier 2, France
Antonio Robles-Gómez, University for Distance Education, Spain
Ateet Bhalla, Ateet Bhalla, NRI Institute of Information Science, India
Aurelio Bermudez, Universidad De Castilla-la Mancha, Spain
Baoying Wang, Waynesburg University, USA
Bastian Koller, Universität Stuttgart, Germany
Blanca Caminero, Universidad de Castilla-La Mancha, Spain
Carlos Molinero, Universidad Complutense De Madrid, Spain
Carlos Duarte, University Of Lisbon, Portugal
Carmen Carrion, University Of Castilla-la Mancha, Spain
Carmen Ruiz, Universidad De Castilla-la Mancha, Spain
Cecile Favre, Université Lyon 2, France
Cesar Andres, Universidad Complutense De Madrid, Spain
Dick Stenmark, Göteborg University, Sweden
Dimosthenis Kyriazis, National Technical University Of Athens, Greece
Djamila Ouelhadj, University Of Nottingham, United Kingdom
Elias Xidias, University Of Patras, Greece
Enrique Árias, Universidad De Castilla-la Mancha, Spain
Federico Bergenti, University of Parma, Italy
Fei Luo, East China University Of Science & Technology, China

Copyrights

Filippos Azariadis, University Of The Aegean, Greece
Francesca Lonetti, Isti-cnr, Italy
Francisco Garcia, University Of Salamanca, Spain
Giacomo Cabri, University of Modena and Reggio Emilia, Italy
Gilles Hubert, Université Paul Sabatier, France
Grigorios Beligiannis, University Of Western Greece, Greece
Guillaume Cabanac, Université Toulouse 3, France
Hao Wu, Yunnan University, China
Hind Castel, Institut National Des Télécommunications, France
Ivan Jelinek, Czech Technical University, Czech Republic
Jan Krasniewicz, Birmingham City University, United Kingdom
Javier Oliver Villarroya, Technical University Of Valencia, Spain
Jiann-Liang Chen, National Taiwan University of Science and Technolo, Taiwan
Jie Tao, Universität Karlsruhe, Germany
Jo Abrantes, University Of Wollongong, Australia
Johannes Meinecke, SAP AG, Germany
Jose Manuel Molina, Universidad Carlos Iii De Madrid, Spain
Jose Santa, University of Murcia, Spain
Juan J. Pardo, Universidad De Castilla-la-mancha, Spain
Juan José Sánchez Peña, Universidad De Alcalá, Spain
Julio Calvo, Ciemat, Spain
Konstantinos Giotopoulos, University Of Patras,, Greece
Konstantinos Tserpes, National Technical University Of Athens, Greece
Kuan-Ching Li, Providence University, Taiwan
Kuo-chan Huang, National Taichung University, Taiwan
Luca Anselma, University Of Torino, Italy
Luciano Senger, State University Of Ponta Grossa, Brazil
Manuel Gil Pérez, University of Murcia, Spain
Marcelo Ponciano-Silva, Universidade De São Paulo, Brazil
Marcio De Souza, Universidade Estadual De Ponta Grossa, Brazil
Marco Botta, University Of Torino, Italy
Marcos Quiles, Federal University of Sao Paulo, Brazil
Marek Woda, Wroclaw University Of Technology, Poland
Maria Camila Barioni, Universidade Federal Do Abc, Brazil
María N Moreno García, Universidad De Salamanca, Spain
Martin Fredriksson, Blekinge Institute of Technology, Sweden
Matthias Lange, Leibniz Institute Of Plant Genetics And Crop Plant, Germany
Max Chevalier, Université De Toulouse, IRIT, France
Mehmet Sahinkaya, University Of Bath, United Kingdom
Mei-Ling Shyu, University Of Miami, USA
Michael Vrahatis, University Of Patras, Greece
Min-Ling Zhang, Hohai University, China
Miroslav Bures, Czech Technical University in Prague, Czech Republic
Moschopoulos Charalampos, University Of Patras, Greece
Nataliya Rassadko, Università Degli Studi Di Trento, Italy
Nikolaos Doulamis, National Technical University Of Athens, Greece

Copy Rights

Nikolaos Matsatsinis, Technical University Of Crete, Greece
Nikolaos Sapidis, University Of The Aegean, Greece
Olivier Teste, Université Paul Sabatier, France
Patrice C. Roy, Université de Sherbrooke, Canada
Pedro Henrique Bugatti, University Of São Paulo, Brazil
Philipp Wieder, Technical University Of Dortmund, Germany
Pierre Busnel, Université de Sherbrooke , Canada
Qin Ding, East Carolina University, USA
Rafa Al-Qutaish, Al Ain University of Science & Technology, UAE
Rafael Casado, University Of Castilla-La Mancha, Spain
Rami Yared, Japan Advanced Institute Of Science And Technology, Japan
Renato Ishii, Universidade Federal de Mato Grosso Do Sul, Brazil
Riad Mokadem, Paul Sabatier University, France
Ricardo Fernández, Universidad De Murcia, Spain
Rodrigo Cilla, Universidad Carlos III De Madrid, Spain
Roland Kuebert, University Of Stuttgart, Germany
Ronaldo Prati, Universidade Federal do ABC, Brazil
Salima Benbernou, Université Claude Bernard Lyon 1, France
Sharon Cox, Birmingham City University, United Kingdom
Shu-Ching Chen, Florida International University, USA
Simon Richir, Arts et Metiers ParisTech, France
Spiridon Likothanassis, University Of Patras, Greece
Stephane Maag, Telecom & Management Sudparis, France
Stephanos Mavromoustakos, European University Cyprus - School of Sciences, Cyprus
Tudor -Razvan Niculiu, University "Politehnica" Bucuresti, Romania
Vasilis Delis, Research Academic Computer Technology Institute, Greece
Vassiliki Andronikou, National Technical University Of Athens, Greece
Victor Robles, Technical University Of Madrid, Spain
Vincenzo Deufemia, Università di Salerno, Italy
Wenbin Jiang, Huazhong University Of Science And Technology, China
Yazid Benazzouz, Orange Labs - France Telecom, France
Yijiao Yu, Central China Normal University, China
Yoshifumi Manabe, NTT Communication Science Laboratories, Japan
Zaher Aghbari, University Of Sharjah, United Arab Emirates

Copy Rights

Copy Rights

KEYNOTE LECTURES

ESKILLS – A CHALLENGE FOR THE MODERN SOCIETY

By Professor Vasile Baltac
ATIC - IT&C Association of Romania, Vice-Chairman of World Information Technology and Software Alliance (WITSA) and President of the Council of European Professional and Informatics Societies (CEPIS), Romania

Abstract

The paper will discuss why eSkills are important, the eSkills gap, the importance of IT professionalism, possible standards for professionalism, the impact of User eSkills and business – university cooperation in ICT.

Information/Knowledge Society needs new skills, the demand has grown rapidly both at pratitioner level and user level. This demand is quantified as a proposed law, a corollary to Moore’s Law. The acceleration of innovation in ICT is ananlysed both with accelerating and decelerating factors. The complexity issue is considered within the debated singularity and Internet’s Omega Point.

Applications are considered a key challenge, as they appear and are implemented at a much reduced accelerated speed.

eSkills Gap is one of the issues related to advance of ICT. Europe faces shortages and their reasons are discussed. Solutions to make ICT careers more attractive to young people both males and females are a key issue for Europe.

Professionalism, professionalism in IT, validation of professionals and researchers are further discussed in line with CEPIS Vision on professionalism.

eInclusion with its upcoming second digital divide can be a major factor of limitation of ICT spread.

Copy Rights

AN INTEGRATED PORTAL FRAMEWORK FOR PROVIDING WEB SERVICES AND E-LEARNING FACILITIES

By Dr. Alina Andreica,
Associate Professor, Head of IT Department,
Babes-Bolyai University, Cluj-Napoca, Romania

Abstract

We describe means of creating an integrated web portal framework for providing e-learning services and dedicated information systems facilities. The portal uses MS technology and provides, as learning services, management content and e-learning facilities for various user categories, together with the dedicated information system facilities. The information system facilities are provided into the web portal by retrieving the dedicated software services from the specific systems and synchronizing databases based on various technologies (php / postgresql, asp / MS sql). This web framework has a good extensibility degree and may be used in order to integrate web services for content sharing and communication, as well as dedicated information system facilities, in various organization cases

Copy Rights

Full Papers

Copy Rights

Copy Rights

FUNCTIONAL TESTING CRITERIA BASED ON FEATURE MODELING FOR SOFTWARE PRODUCT LINE

Danilo Modesto de Sousa and Marcelo Fantinato
School of Arts, Sciences and Humanities (EACH) – University of São Paulo (USP)
Rua Arlindo Béttio, 1000 – 03828-000, São Paulo – SP, Brazil

ABSTRACT

Product line (PL) is an approach for the development of similar products with increasing use in the software industry, and feature modeling is one of the techniques used to represent the product families' variabilities and commonalities. This paper introduces a set of criteria for functional testing of software product lines based on feature models. Such criteria are particularly useful for the business process management domain with Web services as the PL target software. The results of an experiment to perform a comparison among the criteria and to validate their using feasibility are also presented.

KEYWORDS

Software testing, Testing criteria, Feature modeling, Product line, Business Process Management (BPM).

Copy Rights

1. INTRODUCTION

Software testing can be guided by system models that provide information to support test case development and coverage analysis. Depending on the model nature, testing can be: functional – guided by the external specification with the aim of exercising different elements of such specification when running tests on the executable code; or, structural – guided by the internal structure of the generated source code with the aim of exercising different elements of such code (Pressman, 2004). The main objective is to detect defects as early as possible during the software development. Since exhaustive testing is unfeasible, testing criteria are used to assist in the selection of good test cases, i.e. those more likely to find defects (Myers, 1979).

Product Line (PL) is a software engineering approach that promotes the generation of specific products, from a product family, based on the reuse of a core infrastructure and a well defined set of components (SEI, 2007). Feature modeling is an essential technique for capturing and managing commonalities and variabilities that can be applied in PL (Kang, et al., 1990; Czarnecki, et al., 2005). An important application domain for PL is the Business Process Management (BPM), which includes activities that enable the modeling, execution and analysis of interorganizational business processes (Fantinato, et al., 2010). BPM is commonly supported by service-oriented computing, mainly through Web services technology.

Companies are increasingly adhering to the PL approach, mainly due to economic reasons because of its great support for reuse (Linden, et al., 2007). An important concern is the quality assurance of the software produced by the PL which requires an improvement in test strategies for of this type of approach. This paper proposes a set of criteria for functional software testing, based on feature models, which can help in choosing a subset of more efficient test cases for PL, and can be applied for the BPM domain taking into account that the PL target software are Web services. The objective is providing a more systematic way to select test cases when applying the PL approach, especially when resources to be used in testing are scarce.

This paper covers the following subjects in its sections: PL and feature modeling concepts; the proposed testing criteria; the undertaken experimental evaluation; related work; and, the conclusion and future works.

2. SOFTWARE PRODUCT LINE AND FEATURE MODELING

A PL involves a set of similar applications, in a domain, which can be developed from a common generic architecture and a set of components to populate it. This aims at identifying common aspects and differences among the software artifacts during the PL development process, in order to clarify the decision points in which the adaptation of components to generate specific products can be undertaken (Clements & Northrop, 2001). The PL engineering has two life cycles (Linden, et al., 2007; Clements & Northrop, 2001): domain engineering – involves developing the nucleus of reusable artifacts of the PL, including the PL architecture and software components; and, application engineering – involves developing specific products, through the instantiation of the PL architecture developed during domain engineering.

One way to represent common and variable points, during domain engineering, and then select the desired properties, during application engineering, is the feature modeling technique. A feature is a system property, relevant to some entity, used to capture common characteristics or differentiate the systems in a product family (Czarnecki & Antkiewicz, 2005). Features can be mandatory, optional or alternative. A feature model contains a set of interrelated features and is represented graphically by a tree, where the root represents a major concept and the descendant nodes represent the children features.

A feature model example is presented in Figure 1. The notation used is proposed by Czarnecki, et al., (2005), whose metamodel is presented in Section 2.1. The example presents two basic services provided by an E-Shop: Payment and Shipping. Payment can be done by Credit Card or Debit Card, and Fraud Detection can be optionally used. Shipping may be by Land, air, Sea or any combination of these options.

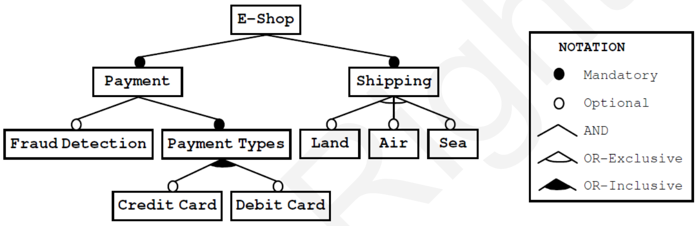


Figure 1. Feature model example – virtual store (Antkiewicz & Czarnecki, 2004)

A feature model describes the configuration space of a product family. A family member can be specified by selecting the desired features from the feature model, considering the variability constraints defined by the model (as the choice of exactly one feature from a set of alternative ones). An example of configuration for the model presented in Figure 1 is: choosing Credit Card for Payment, without fraud detection, and combining Land and Sea for Shipping. This process is called feature configuration (Czarnecki, et al., 2005).

Considering the BPM domain for PL, the features in the model could represent the electronic services being provided by the organizations involved in an electronic negotiation interested in create a cooperative and interorganizational business process. Therefore, each partner organization could elaborate its own feature model, with the features representing structurally the electronic services to be contracted by the other party, which will form the business process in the end in form of Web services (Fantinato, et al., 2010).

2.1 Cardinality-based Feature Metamodel

The cardinality–based feature metamodel, proposed by Czarnecki, et al., (2005), involves the concepts of attributes, feature groups, diagram modularization, and cardinalities. The metamodel is presented as a class diagram in Figure 2. There are three types of features in this feature model: (i) *Root Feature* that forms the root of the different feature diagrams in a model; (ii) *Grouped Feature* that can only occur in a *Feature Group*; and, (iii) *Solitary Feature* that is, by definition, not a *Root Feature* not grouped in a *Feature Group*.

Features can have an *Attribute* with a *TypedValue – String Value* or *Integer Value*. The abstract classes *ContainableByFG* and *ContainableByF* stand for those types of objects that can be contained by a *Feature Group* and a *Feature*, respectively. A *Feature Group* contains *Grouped Features* or *FDReferences*, whereas a *Feature* can include *Solitary Features*, *Feature Groups* and *FDReferences*. Diagram modularization is achieved by using the *FDReference* class, which stands for a feature diagram reference. It can refer to only one *Root Feature*, but a *Root Feature* can be referred by several *FDReferences*.

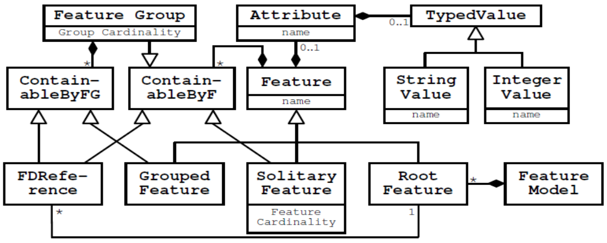


Figure 2. Cardinality-based feature metamodel (Czarnecki, et al., 2005)

Feature and *Feature Group* cardinalities are represented as attributes in the feature metamodel. *Feature Cardinality* defines how often a solitary sub-feature (and possible sub-trees) can be cloned as a child of its parent feature. Similarly, *Group Cardinality* is a property of the relationship between a parent and a set of sub-features. A *Feature Group* expresses a choice over the *Grouped Features* in the group.

The following concepts are defined to be used in the next section for the definition to the testing criteria:

- **Mandatory feature**: is a feature with cardinality [1..1];
- **Optional feature**: is a solitary feature with cardinality [0..1];
- **Alternative feature**: is a grouped feature with cardinality [0..1];
- **Optative feature**: is a optional feature or an alternative feature;
- **Leaf feature**: is a feature with no child feature;
- **Feature level**: represents the distance of a feature in relation to the root feature. For example, for a feature in the level “4”, the root feature is the fourth feature in its ascendency level;
- **Feature level in a group**: represents the distance of a grouped feature in relation to the root feature of the group to which it belongs.

3. FUNCTIONAL TESTING CRITERIA BASED ON FEATURE MODELs

In this section, the functional testing criteria based on feature models are presented. They are proposed to help in selecting and evaluating test cases when applying PL. A testing criterion here defines a set of required elements of a feature model, commonly a specific type of feature, which must be covered by the test cases.

Given FM, a feature model, and C, a set of test cases, the set of testing criteria based on feature models is:

- **Criteria by feature type**:
- **all-features**: requires that all features belonging to the feature model FM associated with the PL are exercised at least once by the set of test cases C;
- **all-mandatory-features**: requires that all mandatory features belonging to the feature model FM associated with the PL are exercised at least once by the set of test cases C;
- **all-optative-features**: requires that all optative (optional or alternative) features belonging to the feature model FM associated with the PL are exercised at least once by the set of test cases C;
- **all-grouped-features-in-level-1**: requires that all grouped features of the first level of a group belonging to the feature model FM associated with the PL are exercised at least once by the set of test cases C;
- **all-leaf-features**: requires that all leaf features belonging to the feature model FM associated with the PL are exercised at least once by the set of test cases C.
- **Criteria by tree level**:
- **all-features-in-level-1-N (N $\geq$ 1)**: requires that all features belonging to the feature model FM associated with the PL, located from level 1 to level N, are exercised at least once by the set of test cases C. It represents a “set of similar criteria” as the value N can refer to any level greater or equal to 1, such as 1, 2, 3, 4 etc; so that the greater the N value, the greater its coverage.

All these criteria are generic regarding the PL life cycle, since they can be applied for both domain engineering and application engineering. Accordingly, they apply to both feature models in general and their configurations. Specifically in an experiment to validate this work, presented in the next section, the proposed criteria were applied in application engineering, i.e. on feature model configurations.

Copy Rights

4. EXPERIMENTAL EVALUATION

An experimental evaluation was carried out to verify the feasibility of applying the proposed criteria and perform a comparison among them. In brief, the experiment was composed by the application of the defined testing criteria on a feature-based PL. The criteria were applied during the PL application engineering, i.e. the test cases were selected according to the feature model configurations. In the following sections, the environment, steps, artifacts, results and analysis of results of the experiment are presented.

4.1 Environment and Context of the Experiment

Ideally, this experiment should have been conducted in an environment composed of a real PL, necessarily based on feature models, consisting of the PL architecture and its components, including a set of already existing test cases – associated with features of the feature model. However, creating such a realistic environment to conduct this experiment was considered impracticable to be used only as a proof of concept. In addition, there were found no real PL ready to be used in the experiment, since the application of this approach is justified only in very complex areas of the software industry.

Thus, this experiment was conducted on a fictional environment, with artificial PL and test cases. First, a dummy feature model was developed, with features created semi-randomly; despite the randomness, the feature model was structured following elements well defined, in order to represent an organized PL. After, fictitious test cases were created and associated to the features, and some of them were defined as test cases whose execution detects the presence of a defect in the software; all the three actions performed randomly.

4.2 Experiment Steps

The following steps were undertaken to conduct this experiment:

1) Establishment of the execution environment, as presented in Section 4.1, including: (a) elaboration of the feature model; (b) elaboration of the test cases; (c) association of features to test cases; and, (d) association of defects to test cases;

2) Derivation of the feature model configurations. Three configurations were generated, with different sizes, so that more comparative data were produced: (a) small – about 40% of features selected; (b) medium – about 50% of features selected; and, (c) large – about 70% of features selected;

3) Selection of the test cases for each configuration, using – for each one – all the testing criteria proposed in Section 3. The criteria **all-features-in-level-1-N** was executed with m N=1, 2 and 3.

Copyrights

4.3 Produced Artifacts

The feature model was elaborated using the FeaturePlugin tool (Antkiewicz & Czarnecki, 2004). Some model excerpts are presented in Figure 3. Several features are presented in their collapsed form. Three views, (a), (b) and (c), are used to present different features expanded and hence different parts of the model.

The PL representation through the feature model was split in two parts: architecture and components. For the PL architecture, eight architectural (components) parts were created. Regular components were classified in three types: (i) mandatory – for which six components were created; (ii) optional – for which five components were created; and, (iii) alternative – for which two component groups were created, one with three alternative components and another one with two alternative components.

For each PL component, a set of features was created to represent the functionalities that compose it. Regardless of these components are mandatory, or optional/alternative, their associated features can be either mandatory or optional/alternatives. Thus, an optional component, for example, may have a mandatory feature associated to it. The total number of features for this feature model, by feature type, is presented in Table 1.

1821 test cases were created and associated to the 350 features, at a rate of 5.2 test cases per feature. 300 defects were associated to 550 of the 1821 test cases; in a way that one defect could have its presence detected by the execution of two or more test cases, although most of the defects were associated with only one test case. Table 2 presents the number of defects associated to the features of the three configurations.

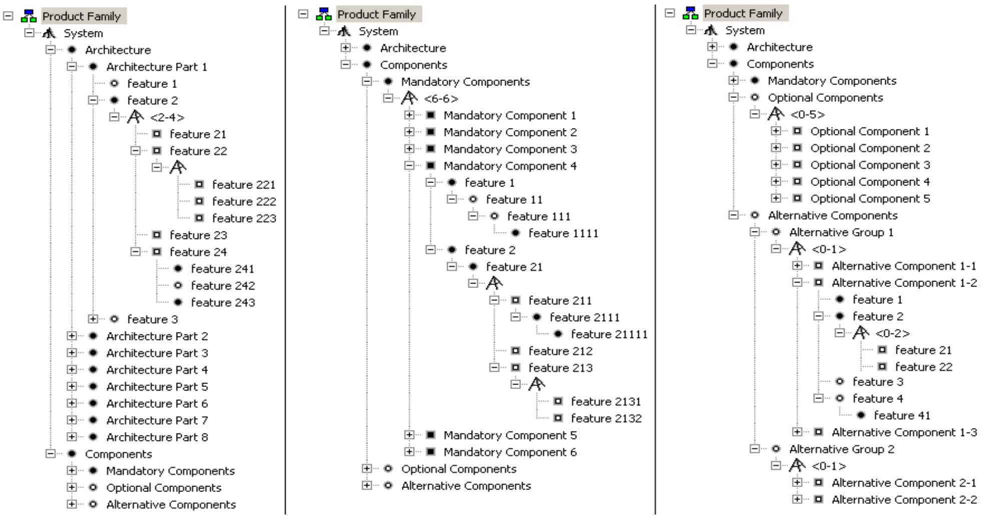


Figure 3. Excerpt of the feature model used in the experiment

Table 1. Number of features used in the experiment

Feature type	Feature model	Feature model configurations		
		Small	Medium	Large
Mandatory	138	79	104	115
Optative (Optional + Alternative)	212	58	78	60
Total	350	136	180	247

Table 2. Number of defects whose presence was detectable by the test cases

Defect types	Feature model configurations		
	Small	Medium	Large
Unique defects	159	196	242
Total defects	200	283	371

Copy Rights

4.4 Achieved Results

This section presents the data obtained by applying the testing criteria in the selection of test cases for testing the products generated from the PL of this experiment. Tables 3 and 4 present data for the five criteria by “feature type”. First, in Table 3, the numbers of required elements are presented. For each criterion and configuration, the following data are presented: the absolute number of required features and the percentage over the total number of existing features in the respective configuration. For example, the feature model configuration 2 (medium) had, as a total, 180 features; for the criterion **all-optative-features**, there were 78 features of this type that should be exercised, which represent 43% of the total value (i.e. of the 180 features).

Table 4 presents the numbers of defects whose presence was detected, according to the random association between test cases and defects performed in the step 1. Two numbers are presented: unique defects – in which different test cases that detect the presence of a same defect are counted only once; and, total defects – in which all the defects whose presence were detected are counted regardless being repeated or not. The maximum numbers of defects (unique and total) whose presence could be detected are presented for comparison. For example, for the configuration 2, a maximum of 196 (unique) and 283 (total) defects could have had their presence detected during test execution; for the criterion **all-optative-features**, 26 e 27 defects (unique and total, respectively) had their presence detected, 13% and 10% of the maximum possible.

Tables 5 and 6 present similar data for the “tree level” criteria. Table 5 presents data regarding the required elements, in which N=3 refers to the maximum level and, therefore, to the total number of features. Table 6 presents data regarding the defects detected by the test cases selected by each criterion.

Table 3. Number of required elements (specific feature types) for each test criterion "by feature type"

		Configuration 1	Configuration 2	Configuration 3	Average
all-features	#	136	180	247	188
	%	100%	100%	100%	100%
all-mandatory-features	#	79	104	115	99
	%	58%	58%	47%	53%
all-optative-features	#	58	78	117	84
	%	43%	43%	47%	45%
all-grouped-features-in-level-1	#	28	38	60	42
	%	21%	21%	24%	22%
all-leaf-features	#	81	99	151	110
	%	60%	55%	61%	59%
Average	#	76	100	138	105
	%	56%	55%	56%	56%

Table 4. Number of defects – test criteria "by feature type"

		Configuration 1		Configuration 2		Configuration 3		Average	
		Unique	Total	Unique	Total	Unique	Total	Unique	Total
Maximum defects		159	200	196	283	242	371	199	285
all-features	#	36	37	48	52	69	82	51	57
	%	23%	19%	24%	18%	29%	22%	26%	20%
all-mandatory-features	#	23	24	26	28	31	34	27	29
	%	14%	12%	13%	10%	19%	9%	13%	10%
all-optative-features	#	17	17	26	27	37	39	27	28
	%	11%	9%	13%	10%	15%	11%	13%	10%
all-grouped-features-in-level-1	#	13	13	16	16	27	28	19	19
	%	8%	7%	8%	6%	11%	8%	9%	7%
all-leaf-features	#	19	20	24	29	26	28	23	26
	%	12%	10%	12%	10%	11%	8%	12%	9%
Average	#	22	22	28	30	38	42	29	32
	%	14%	11%	14%	11%	16%	11%	15%	11%

Table 5. Number of required elements (specific feature types) for each test criterion "by tree level"

		Configuration 1	Configuration 2	Configuration 3	Average
Total features		136	180	247	188
all-features-in-level-1-1	#	45	50	59	51
	%	33%	28%	24%	27%
all-features-in-level-1-2	#	110	107	141	119
	%	81%	59%	57%	64%
all-features-in-level-1-3	#	124	150	192	155
	%	91%	83%	78%	83%
Average	#	93	102	131	109
	%	68%	57%	53%	58%

Table 6. Number of defects – test criteria "by tree level"

		Configuration 1		Configuration 2		Configuration 3		Average	
		Unique	Total	Unique	Total	Unique	Total	Unique	Total
Maximum defects		159	200	196	283	242	371	199	285
all-features-in-level-1-1	#	10	10	13	13	19	21	14	15
	%	6%	5%	7%	5%	8%	6%	7%	5%
all-features-in-level-1-2	#	24	27	20	21	43	44	29	31
	%	15%	14%	10%	7%	18%	12%	15%	11%
all-features-in-level-1-3	#	37	39	39	45	59	61	45	48
	%	23%	20%	20%	16%	24%	16%	23%	17%
Average	#	24	25	24	6	40	42	29	31
	%	15%	13%	12%	9%	17%	11%	15%	11%

Copy Rights

4.5 Results Analysis

The results presented in Section 4.4 showed that different testing criteria present different coverage levels: through Tables 3 and 5, it is possible to verify that the number of required elements by each criterion is different among them. The same applies to de number of defects with presence detected (Tables 4 and 6).

Regarding the required elements for the criteria by feature type (Table 3), **all-features** presented the greatest number of elements to be exercised, since it requires all the existing features. **all-mandatory-features**, **all-optative-features** and **all-leaf-features** had a number of required elements about to 50%, ranging between 40% and 60%; depending on the model configuration, they alternated between a lowest and a highest number. And **all-grouped-features-in-level-1** presented the lowest number of required elements. For the criteria by tree level (Table 5), the results were achieved in accordance with what might be expected: they naturally lead to the criterion in level **N=3** to have more required elements than **N=2** which in turn has more required elements than **N=1**. A similar analysis was performed regarding the number of defects whose presence was detected by each criterion (Tables 4 and 6). In this case, it was expected that the greater the number of required elements for a particular criterion, the greater would be the number of defects found.

Analyzing the criteria and creating a rating according to the average values presented by the three configurations, the data of Table 7 were obtained. Both ratings follow the same pattern, with minor changes in order. A gap in coverage between the criteria was observed: the most stringent criterion (100%) required nearly five times more features than the least stringent one (22%); whereas the most efficient criterion (26%) detected the presence of almost four times more defects than the least efficient one (7%).

This criteria rating analysis could have been done also by mathematical studies, not only experiments. Unfortunately, this does not seem possible for the feature modeling case, since by its formation rule, all features in a model may be exclusively either mandatory or optional ones, for example; or else all features could be located in level N=2 excepted by one feature in level N=1. Although possible, these special usages of feature models are not common in practice, since there are no real-world situations representable by them.

Table 7. Criteria rating regarding coverage of required elements and percentage of defects found

Coverage of required elements			Percentage of defects found		
Position	Criterion	Coverage	Position	Criterion	% of defects
1	all-features	100%	1	all-features	26%
2	all-features-in-level-1-3	83%	2	all-features-in-level-1-3	23%
3	all-features-in-level-1-2	64%	3	all-features-in-level-1-2	15%
4	all-leaf-features	59%	4	all-mandatory-features	13%
5	all-mandatory-features	53%	5	all-optative-features	13%
6	all-optative-features	45%	6	all-leaf-features	12%
7	all-features-in-level-1-1	27%	7	all-grouped-features-in-level-1	9%
8	all-grouped-features-in-level-1	22%	8	all-features-in-level-1-1	7%

5. RELATED WORK

There is a debate about the best strategy for software testing in PL (McGregor, 2007). Four main strategies can be currently found in literature (Tevanlinna, et al., 2004): (i) the individual and independent test of each product generated by the application engineering; (ii) the incremental testing of the PL; (iii) the instantiation of reusable tests; and, (iv) the division of responsibilities.

ScenTED (*Scenario-based TEst case Derivation*) is an example of technique used to instantiate reusable test cases (Metzger, 2006). It supports the creation of domain testing models (as extended UML activity diagrams) by extending use cases models in the requirements level, and the derivation of domain test cases from these models. As a result, the domain test cases are used to derive test cases for a specific application.

According to White, et al., (2008), since feature models are widely used to describe variabilities in PL, they can be used to improve testing in PL, as exploited in the paper. A variation of this idea was presented by Olimpiew and Gomaa (2008), which uses a combination of UML diagrams and feature models: test cases are created based on use cases and activity diagrams; and, through feature models, each feature is associated with one or more test cases. Thus, when a feature is selected, its respective test cases are selected as well.

Copy Rights

6. CONCLUSION

The search for increasing efficiency in software engineering has led the industry to apply approaches related to software reuse. Typically, these approaches are associated only to the first phases of software development – such as analysis, design and implementation. LP is one of these approaches, which aims at systematizing the development of similar software products. Although software testing is an activity whose supporting techniques are usually based on models, few studies have reported testing associated to feature modeling. Feature modeling technique, one of the forms used to represent variabilities and commonalities in product families of PL, has a great potential to provide support in the testing activity. This paper helps to demonstrate this potential, by presenting an experiment in which some proposed testing criteria were used and compared.

The experiment showed the criteria feasibility for use in practice. Moreover, it showed that the criteria have different demanding levels in terms of required elements and consequently in terms of efficiency or, inversely proportional, in terms of resources needed. Thus, test analysts who work with PL based on feature models can have a range of testing selection and coverage criteria. These criteria may be used at different times depending on the objectives and available resources. The experimental results presented were obtained based on an artificial environment. Ideally, a real PL environment, for example for BPM domain, should have been used, so that there was a greater reliability on the reached conclusions. This is a major concern of future research. Another interest for future is the development of a supporting tool for software testing based on the criteria proposed here, both for the selection of test cases and for the coverage analysis.

ACKNOWLEDGEMENT

This work was supported by The State of São Paulo Research Foundation (FAPESP), Brazil.

Copy Rights

REFERENCES

Antkiewicz, M. and Czarnecki, K., 2004. FeaturePlugin: Feature Modeling Plug-in for Eclipse. *Proceedings of eTX workshop*. Vancouver, Canada, pp. 67-72.

Clements, P. and Northrop, L., 2001. *Software Product Lines: Practices and Patterns*. SEI Series in Software Engineering, Addison-Wesley, New York, USA.

Czarnecki, K. et al, 2005. Staged Configuration Through Specialization and Multi-level Configuration of Feature Models. *In Software Process: Improvement and Practice*, Vol. 10, No. 2, pp. 143-169.

Czarnecki, K. and Antkiewicz, M., 2005. Mapping Features to Models: A Template Approach Based on Superimposed Variants. *Proceedings of 4th International Conference on Generative Programming and Component Engineering*. Tallinn, Estonia, pp. 422-437.

Fantinato, M. et al, 2010. Product Line in the Business Process Management Domain. *In: K. C. Kang, V. Sugumaran, S. Park (Eds.), Applied Software Product Line Engineering*, Auerbach Publications, Boca Raton-FL, USA, pp. 497-530.

Kang, K. et al, 1990. *Feature-Oriented Domain Analysis (FODA) Feasibility Study*. Technical Report CMU/SEI-90-TR-021, SEI/CMU, USA.

Linden, F. J. et al, 2007. *Software Product Lines in Action: The Best Industrial Practice in Product Line Engineering*. Springer, Berlin, Germany.

McGregor, J. D., 2008. Toward a Fault Model for Software Product Lines. *Proceedings of 5th Software Product Lines Testing Workshop*. Limerick, Ireland, pp. 157-162.

Metzger, A., 2006. Model-based Testing of Software Product Lines. *Proceedings of 7th International Conference on Software Testing*. Duesseldorf, Germany.

Myers, G., 1979. *The Art of Software Testing*. John Wiley & Sons, Hoboken, New Jersey, USA.

Tevanlinna, A. et al, 2004. Product Family Testing: A Survey. *In ACM SIGSOFT Soft. Eng. N.*, Vol. 29, No. 2, pp. 12-12.

Pressman, R., 2004. *Software Engineering: A Practitioner's Approach*. Mc-Graw Hill, New York, USA.

SEI – Software Engineering Institute, 2007. *A Framework for Software Product Line Practice - Version 4.2*, http://www.sei.cmu.edu/productlines/framework.html.

White, J. et al, 2008. Automated Diagnosis of Product-line Configuration Errors in Feature Models. *Proceedings of 12th International Software Product Line Conference*. Limerick, Ireland, pp. 225-234.

A NOVEL METHODOLOGY TO FORMALIZE THE REQUIREMENTS ENGINEERING PROCESS WITH THE USE OF NATURAL LANGUAGE

Marinos G. Georgiades* and Andreas S. Andreou**
**University of Cyprus*
Department of Computer Science, P.O.Box 20537, 1678 Nicosia, Cyprus
***Cyprus University of Technology*
Department of Electr. Engineering and Information Technology, P.O.Box 50329, 3603 Limassol, Cyprus

ABSTRACT

The lack of a formal approach with high expressiveness close to that of natural language drives systems analysts to use their own informal ways to engineer requirements. This paper describes a novel methodology that attempts to formalize a large part of the Requirements Engineering (RE) process, including Discovery, Analysis and Specification of requirements. The formalization is achieved by utilizing elements of natural language syntax and semantics, with the focus being on keeping ambiguities low and expressiveness high. In particular, Requirements Engineering is converted to a series of predefined steps, through which the analyst is guided in advance what specific types of data and functions to use, how to form and document them, and, more importantly, what (predefined) questions to ask the stakeholders in order to correctly elicit their needs. The proposed methodology can take an object-oriented or a functional direction. It is supported by a software tool, which also offers automatic construction of diagrammatical representations.

KEYWORDS

Natural Language, Automated Requirements Engineering.

Copy Right

1. INTRODUCTION

Recent studies show that the least understood parts of systems' development are the stages of requirements discovery, analysis and specification (The Standish group, 2009). The problem observed is that there is an enormous gap between the clients' needs and the software engineers' understanding of the clients' needs (Goldin, 1997). Clients often speak with vague sentences and/or cannot express their functional needs or, even worse, they do not know what these needs really are. This problem is amplified further when the analyst does not provide the right questions, as he/she essentially does not know precisely what to ask.

Our standpoint is that if you know what to write, then you know what to ask. Hence, if the analysts know, in advance, specifically what types of functions, data and constraints (Requirements Analysis - RA) they should search for and write down, then they will be able to ask specific questions (Requirements Discovery - RD) regarding that particular information. A second priority of engineering the requirements is to formalise the way the analysts write this information (Requirements Specification - RS) - that is, to organize it, apply correct syntax, etc. Similarly, the way the RD questions are written is part of this (second) priority. Conclusively, building the questions for RD, based on RA (mainly) and RS is a reliable way to derive the right answers/requirements from the users.

Such an approach or methodology that provides specific steps in advance and, more importantly, a formalized and understandable way to engineer requirements does not currently exist. This paper proposes the NLSSRE (Natural Language Syntax and Semantics RE) methodology, based on prior work by Georgiades et al. (2005), that utilizes elements of natural language (NL), such as verbs, nouns, genitive case, adjectives, and adverbs, to formalize the stages of RD, RA and RS. The main concept of the methodology is that it can formalize the RE process by providing specific types of functions and data, as well as patterns of formalized sentences, for the RA and RS stages, based on which specific pre-determined questions are created to be used in the RA stage. Our decision to adopt NL is based on three significant expectations for this endeavour: (i) to identify and define adequately the various types of data and functions of an information system (IS), as well as their relations, because language, by its nature, is the most powerful medium of expression; (ii) to provide a common terminology and eliminate redundancies in specifying names of functions, data and constraints; and (iii) to give requirements a NL-like description which is very understandable and useful as a communication medium between users, analysts, and programmers of the IS.

This paper is structured as follows: Section 2 reviews the existing literature on the use of NL in RE. Section 3 discusses in detail the underpinning of the methodology, as well as each of its steps with some illustrative examples. Section 4 provides some conclusions and recommendations for future work.

2. LITERATURE REVIEW

The NLSSRE methodology covers the stages of RA, RD and RS. Its key-point is that the RA (mainly) and RS use predefined types and templates of functions and data, and they guide the process of building the question-sets for the RD stage; and the answers to these questions complete RA and RS by giving values to the relevant types and templates. In this section, we examine other approaches of NL in RE separately for each stage of RE, attempting to provide some sort of comparison with our methodology.

The approach that dominates the literature of RD with the use of NL is the retrieval of requirements from already written requirements documents, by using rule-based (Goldin and Berry, 1997; Rolland and Proix 1992; Li et al., 2005) or probabilistic techniques (Rayson et al, 1999). There are also a few approaches (Tjong et al., 2006; Videira and da Silva, 2005) which suggest that users should write a paragraph describing their job tasks in free text on which they also apply similar retrieval rules to elicit the requirements. However, the retrieval approach is not particularly reliable, since requirements are often not written syntactically, grammatically and semantically correctly from scratch, and the rules applied to retrieve them cannot work well to produce reliable and complete results; additionally, there is a good possibility that the original texts do not cover all the requirements of the IS under development and also include redundancies and disorganized material. In contrast, our approach differs from the aforementioned ones, since it provides specific (predefined) sets of questions, which we derive from the formalization of functions and data. The answers to these questions finalize the analysis and specification stages. Hence, the way we try to elicit requirements is clearly connected to the analysis and specification of requirements. In the current literature, this link does not exist, and this is exactly why the resulting requirements documents need to be re-organized, re-validated and re-adjusted.

In the requirements analysis stage, issues such as identification, classification and decomposition of requirements need to be dealt with. Videira and da Silva (2005) and Rolland and Proix (1992) mention only types of actions based on the verbal types, but they do not expand on them or match them with IS elements. Our approach differs from the aforementioned studies and expands on the use of language semantics to identify, classify, group and decompose IS functions and data (for example, we use genitive case types to define different types of data), to define constraints and non-functional requirements.

Regarding requirements specification, which deals with the construction of the requirements syntax, some approaches (e.g. Conger's, 1994) use a basic syntax (<Subject> <Verb> <Object>), and others use an additional element (<Subject> <Verb> <Object> <Complement>) such as in Rolland and Proix [5]. However, these approaches do not provide specialisations of this basic syntactic form according to the type of the verb (Create, Alter, Erase, etc.) and in turn they cannot group such specialised requirements under one comprehensive function or an object. This is provided in our approach, where we also map the different syntactic forms to either a functional or an object-oriented specification to produce a complete requirements specification document.

3. THE METHODOLOGY

The goal of the methodology is the formalization of the major activities of RE including Requirements Discovery, Analysis and Specification, so that the analyst will know in advance, through a step-by-step approach, what questions to ask, in what specific way to analyse the answers to the questions, and how to write them in a specific way. The application domain of the methodology is an IS (e.g. Hospital IS or

Copyright

Bookstore IS) that deals mainly with management of documents or other physical objects that can be conceived as electronic information which can be Created, Altered, Read and Erased.

3.1 Fundamentals

The formalization of the methodology is based on two elements: Firstly, on the way the methodology is built, and secondly on Natural Language. For the former, we followed our standpoint, as already mentioned in the introduction, that if you know what to write, then you know what to ask. Hence, if we know what functions and data we look for (this is part of RA), then we will know what questions to create (activity of RD), which, in turn, will give specific answers about the said data and functions. As a result, in our methodology, first we use predefined types of functions and data, and based on these we derive the questions for RD. The second element that facilitates formalisation is the use of Natural Language. NL gives expressiveness to the formalization of requirements and makes them easily understood by the users, analysts and programmers. In particular, we use several linguistic elements from semantics and syntax of natural language. In our approach, for the RA stage, data are derived from the semantic types of genitive case, other grammatical cases, nouns, adjectives, adverbial complements, and stable and temporary object properties;[1] functions are derived from the semantic types of verbs; finally, constraints are derived from relations between data and between data and functions. For the RS stage, functions, data and constraints can be written in the form of formalized sentences, by using the right order of different syntactic parts, such as subject, direct object, indirect object, etc., and grouped based on either a functional approach (figure 3a) or an object-oriented approach (figure 3b). We also use the above and other linguistic elements to provide a common terminology for documenting data, functions and constraints. The advantage is twofold: first there will be a consistent and common language of writing, without ambiguities and redundancies, and, second, this controlled language may be computer-processed and translated automatically into semi-formal notations, such as diagrams (UML class diagrams, use case diagrams and specifications, and DFDs), or formal notations, such as Z specifications.

3.1.1 Information Object

Our world consists of either tangible objects (concrete - which we can feel by using our 5 senses, e.g. book, chocolate) or intangible objects (abstract - e.g. disease). In this work *an Information Object (IO) denotes a separate entity of information (attributes) that can stand on its own in the IS. The IO can be created, altered, read and erased within the context of the IS.* For example, a car tyre is an IO since it can be separated from a car and, for example, be used in another car, or a *Doctor* is an IO, since it is a separate entity consisting of a set of attributes, such as height, weight, specialty, etc., while a cup handle is not separable from the cup (when in that case it will have no use), and may not be considered an IO[2].

In the NLSSRE methodology, for each IO, five (5) patterns of formalized sentential requirements (FSRs) are provided, as shown in the example of fig. 2(a) and in section 3.1.3. Each FSR pattern includes the following elements: (a) a CAREN function (**C**reate, **A**lter, **R**ead, **E**rase, **N**otify) which is applied on each IO and also denotes the type of the FSR; (b) non-functional requirements (Instrument, Amount, Time, Location – not elaborated in this paper, due to space limitation) with direct relation to each CAREN function; (c) Roles (e.g. Creator, Accompaniment) that are related to each CAREN function and are also attributes of the IO; and (d) constraints, an example of which is given in section 3.1.3 (not elaborated in this paper, due to space limitation). Hence, the FSRs facilitate the formalization of functions, data attributes, non-functional requirements and constraints of the IO. For the formalization of additional types of attributes of each IO, the NLSSRE methodology makes use of the genitive case, the adjective and other types of attributes. In section 3.1.2 we illustrate the use of the genitive case for the formalization of some types of attributes. In section 3.1.3, we illustrate the use of FSRs to formalize the rest of the aforementioned elements. Additionally, we will show how the functions and data of the system are grouped in relation to the IO, leading either to a functional or an object-oriented specification.

The issue of identifying the IOs - requires the involvement of users and stakeholders - is critical in Requirements Analysis and has not been examined extensively in the relevant literature. It is also the first step of our methodology (section 3.2). The identification of IOs will help us organize the data (IOs and

[1] In this paper, we focus on the use of genitive case – the use of adjectives, which helps derive sub-types of data and functions as well as defining constraints, and the use of adverbial complements for defining constraints will be part of future work.

[2] There are some cases where inseparable parts can be IOs, but expansion on this topic is out of the scope of this paper.

attributes) of the IS and their relationships. However, this issue will not be explored in the current paper, due to space limitations, and will be left for future work.

3.1.2 Formalization of Data Attributes of the IO

We distinguish three types of IO attributes: The Primitive attributes, which are related to the IO per se and usually refer to its physical characteristics (e.g., for the *Patient* IO, primitive attributes include temperature, height, mass), the Peripheral attributes that refer to other IOs related to the IO under study (e.g., for *Patient*, peripheral attributes include *Doctor*, *Disease*), and the Document attributes (e.g. title, fonts, etc.), since each IO can be treated in the form of a document (electronic or paper).

As mentioned in 3.1.1 the IO attributes are determined by the FSR roles, the genitive case, the adjectives, permanent and temporary object properties, and other types of attributes. Here, due to space limitations, we describe only the use of genitive case. The genitive case is the linguistic case that provides relationships between nouns. In this paper, we utilize the relationship types that denote origin, and purpose/use (others include genitive of possession, material, composition, content, and magnitude) to give an indication of the use of genitive case for identifying types of data.

Attributes from Genitive of Origin: The genitive of origin expresses the source, person or place from which something originates. In IS, the genitive of origin corresponds to the Creator(s) of the IO, and so the IO will include *Creator(s)* as attribute(s). For example, let us assume that the *Translation* IO is created by 3 entities: the *Translation Coordinator*, the *Proofreader* and the *Translator*. Hence, these 3 entities will be attributes of the *Translation* IO (they could also be new IOs that will also be analysed for their attributes and functions separately). The *Creator* can also have an Accompaniment who helps him/her during the creation of the IO (e.g. *Translator* is the *Creator* and the other two are *Accompaniments*). Additionally, a *Creator* has responsibility for the creation of the IO, and hence Creator's Signature could be another derived attribute.

Attributes from Genitive of Purpose: It indicates the purpose for which something is used. In particular, the genitive substantive denotes the purpose or intended recipient of the head noun. In an IS, the genitive of purpose corresponds to two attributes of the IO, the Intended recipient who will use the IO according to its intended use (in this context *using* it means changing it or evaluating it, but not just transferring it which has no effect on the object) and the Purpose/ Use of the IO which is the intended use for which it is created.

In summary, according to some indicative types of the genitive case, an IO can have the following attributes: Creator, Signature, Accompaniment, Intended Recipient, Purpose/Use, Owner and Physical attributes such as height, weight, and body parts.

3.1.3 FSRs for the Formalization of Functions, IO Attributes, Non-Functional Requirements and Constraints

Figure 1 below shows **CAREN**, the set of functions and sub-functions we recommend, which are applied on the IO. The formalization of functions is based on "key" types of verbs in natural language grammar that are related to electronic information – taking also into account the relevant RE literature – and can be considered as functions of an IS. **C**reation, **A**lteration, **R**eading, and **E**rasure are applied on an Information Object (IO), **N**otification is applied (triggered) after the Creation, Alteration, or Erasure of an IO, while Addition, Removal and Comparison are lower level functions that are applied on the properties/attributes of each IO. Reading can be also part of Creation, Alteration and Erasure as will be explained later below.

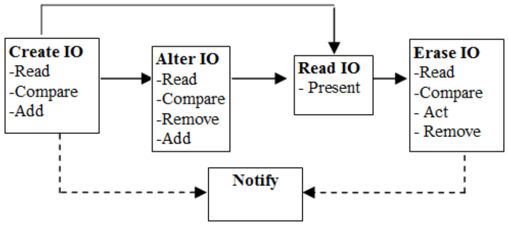


Figure 1. CAREN - A recommended set of functions and sub-functions applied on the IO, and the notifications produced.

Every CAREN function is part of the Formalized Sentential Requirement (FSR) which also includes the relevant IO, roles and non-functional requirements of the IS. The syntax of the FSR helps us organize all these related parts in one sentence. Below we provide the syntax of each FSR pattern, which varies according to each CAREN function, and it is based on the general form <Subject><Verb> <Object><Adverbial Complement>. Additionally, figures 2(a,d) provide examples of the FSRs syntax from a real case.

Create: Creation is the most significant function, since during Creation the attributes of the IO take their initial values which are the basis for further processing by the remaining functions. Creation FSR has the following syntax: <Creator,Accompaniment><Create><IO><Adverbial Complement>::

<System><Notifies><Creator,Accompaniment><Intended Recipient,Notifiee><Adverbial Complement>

where Creator is the entity that creates the IO, Accompaniment is the entity that assists the Creator in the creation of the IO, Intended Recipient is the entity for which the IO is created and which will utilize the IO within the IS, and Notifiee is the entity that needs to be notified of the creation of the IO (this entity will not use the IO in any way that will cause any interaction within the system). On the right of the symbol "::" the syntax of the Notification function follows, which is triggered after the execution of the function on the left. Adverbial complements denote non-functional requirements with the types of *Instrument*, *Time*, *Quantity*, and *Place*. As depicted in figure 2(d), for the Prescription IO, the Creation FSR is written as follows:

<Doctor,Nurse><Create><Prescription><Stylus;11:00a.m.;10minutes;Doctor'soffice)::
<System><Notifies><Doctor,Nurse,Pharmacist,Patient>

Creation, similarly to the other CAREN functions, is decomposed to the sub-functions of Reading, Comparison and Addition. Reading is the sub-function that will present the attributes of the IO to the Creator, Comparison will check if the value to be assigned to an IO attribute satisfies the constraints about that attribute, and Addition will add the approved values.

Alter: During Alteration, the value of one or more of the attributes of the IO changes. The Alteration FSR has the following syntax: <Alterator,Accompaniment><Alter><IO><Adverbial Complement>::

<System><Notifies><Alterator,Creator,Accompaniment><Intended Recipient,Notifiee>

where Alterator is the entity that alters the IO.

Read: The meaning of this function can be conceived in two ways: the first, which is the one that concerns Requirements Analysis, is about what the user wants to read from the data of a particular IO; some of the data need to be provided to the user and some other may not. The second concept for 'Read' concerns the way this data will be presented, including drawings, graphics, video, multimedia, etc; a part of this concept (the general one) falls in RS, but the detailed procedures of implementing such methods of presentation concerns the Design which is outside of the scope of RE. Reading is also embedded and occurs as a first sub-function of the functions of Creation, Alteration and Erasure. Reading has the following syntax:

<Experiencer><Reads><IO><Adverbial Complement>

where Experiencer is the entity that experiences the IO through viewing it, listening to it, etc.

Erase:[3] Erasure of the IO means that the Information Object is permanently deleted. All of its information, including properties and functions, is deleted. Erasure has the following syntax:

<Alterator><Erase><IO>:: <System><Notifies><Alterator,Creator,Notifiee>

Notify: At the user's level, assuming a manual, paper-based IS, we meet the function of Transmission (from the linguistic verb of Transfer of Possession), where data is sent from one entity to another; for example, the Doctor gives the Prescription to the Patient, and the Patient gives the Prescription to the Pharmacist. In a computerised IS the Transmission of Prescription is replaced by the *Read* function, since the IO (prescription, in this case) is already stored (after creation or alteration) in the IS. Hence, the Pharmacist can Read the Prescription by simply retrieving it from the database. However, in a computerized IS, transmission exists at the messaging level, and we name it Notification. In particular, when an IO is created or altered, then a notification should be sent to the interested parties (including Intended Recipients).

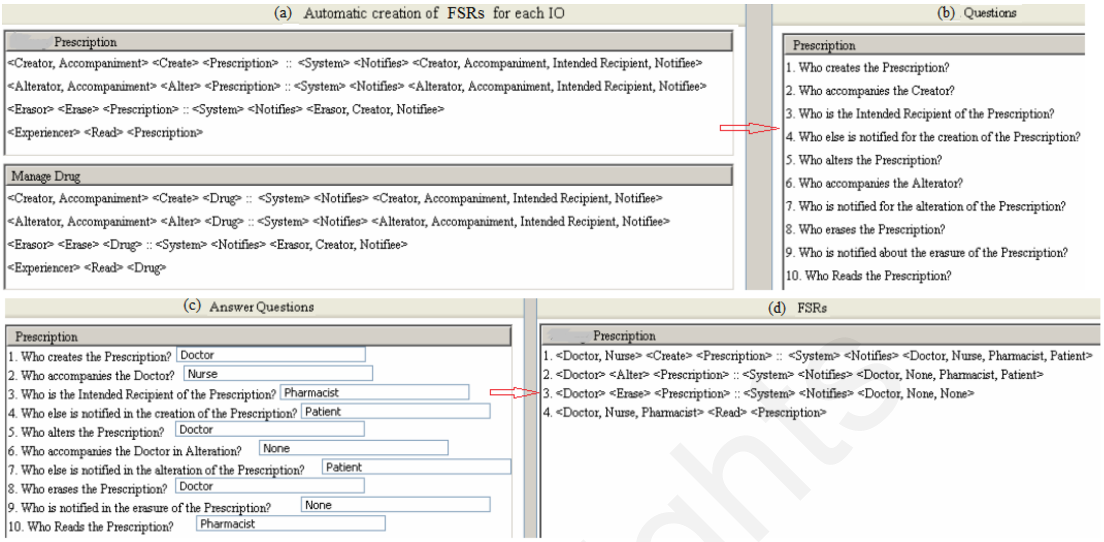
3.1.4 Predefined Question Sets – Elicitation

Most, if not all, of the approaches that use formalism in NL RE try to develop and formalize requirements that are already written in existing documents. We consider this approach not efficient, since requirements in such documents are often poorly written and organized; sentences do not necessarily follow the correct form of syntax, while there may exist redundant words, fuzzy and complicated meanings, etc. As such, it is rather precarious and difficult to apply linguistic rules on such documents.

Based on the syntax of each FSR (as in the example of fig. 2a), relevant predetermined questions are created (fig. 2b) that guide stakeholders to provide specific answers without ambiguities, vagueness and

[3] Since there are interdependent entities, the user must be informed that by erasing an object another object may be affected. Therefore, appropriate constraints should be defined along with relevant questions for the user.

redundancies; these questions are submitted to the stakeholders (fig. 2c), and their answers are used to complete the requirements (consisting of complete FSRs like in fig. 2d, IO attributes, and grouping). Subsequently, the complete formalised requirements are utilised, with the aid of specific rules, to build ORDFDs, class diagrams and use case diagrams and specifications.


(a) Automatic creation of FSRs for each IO

Prescription

<Creator, Accompaniment> <Create> <Prescription> :: <System> <Notifies> <Creator, Accompaniment, Intended Recipient, Notifiee>
<Alterator, Accompaniment> <Alter> <Prescription> :: <System> <Notifies> <Alterator, Accompaniment, Intended Recipient, Notifiee>
<Erasor> <Erase> <Prescription> :: <System> <Notifies> <Erasor, Creator, Notifiee>
<Experiencer> <Read> <Prescription>

Manage Drug

<Creator, Accompaniment> <Create> <Drug> :: <System> <Notifies> <Creator, Accompaniment, Intended Recipient, Notifiee>
<Alterator, Accompaniment> <Alter> <Drug> :: <System> <Notifies> <Alterator, Accompaniment, Intended Recipient, Notifiee>
<Erasor> <Erase> <Drug> :: <System> <Notifies> <Erasor, Creator, Notifiee>
<Experiencer> <Read> <Drug>

(b) Questions

Prescription

1. Who creates the Prescription?
2. Who accompanies the Creator?
3. Who is the Intended Recipient of the Prescription?
4. Who else is notified for the creation of the Prescription?
5. Who alters the Prescription?
6. Who accompanies the Alterator?
7. Who is notified for the alteration of the Prescription?
8. Who erases the Prescription?
9. Who is notified about the erasure of the Prescription?
10. Who Reads the Prescription?

(c) Answer Questions

Prescription

1. Who creates the Prescription? Doctor
2. Who accompanies the Doctor? Nurse
3. Who is the Intended Recipient of the Prescription? Pharmacist
4. Who else is notified in the creation of the Prescription? Patient
5. Who alters the Prescription? Doctor
6. Who accompanies the Doctor in Alteration? None
7. Who else is notified in the alteration of the Prescription? Patient
8. Who erases the Prescription? Doctor
9. Who is notified in the erasure of the Prescription? None
10. Who Reads the Prescription? Pharmacist

(d) FSRs

Prescription

1. <Doctor, Nurse> <Create> <Prescription> :: <System> <Notifies> <Doctor, Nurse, Pharmacist, Patient>
2. <Doctor> <Alter> <Prescription> :: <System> <Notifies> <Doctor, None, Pharmacist, Patient>
3. <Doctor> <Erase> <Prescription> :: <System> <Notifies> <Doctor, None, None>
4. <Doctor, Nurse, Pharmacist> <Read> <Prescription>

Figure 2. The predefined questions (b) created automatically by the FSRs patterns (a), and the resulting FSRs (d) created automatically by the answers of the users (c), for the *Prescription IO* – screenshots are taken from our software tool that implements and supports the proposed methodology.

3.2 Methodology Application Steps

The ultimate goal of the methodology is to be able to apply it in a real setting. The application steps of the methodology are the following:

a. Identify the Information Objects (IOs) of the system. Within this step we use specific guiding rules (as illustrated in section 3.1.1), to distinguish the IOs of the IS. For example, for a Hospital IS, some of the IOs include *Prescription*, *Pharmacy*, *Patient*, and *Doctor*.

b. Identify attributes of each IO. Within this step we identify the types of attributes of each IO. For this, NLSSRE provides predefined types of attributes derived from genitive case types, as explained in section 3.1.2, as well as other linguistic semantic roles. For example, the *Prescription* IO will have a *Creator*, an *Accompaniment*, an *Intended Recipient*, etc.

c. Develop FSRs patterns for each IO. For each IO, the five FSRs patterns are created, as shown in the example of figure 2(a). In this example, non-functional requirements are omitted, as they are still under development and integration with the current tool.

d. Create questions for each IO. For each IO, we derive questions from the elements of the FSRs patterns (fig. 2b) and the pre-defined types of attributes (not depicted), in the example of the *Prescription* IO.

e. Collect the answers to the questions from the stakeholders and complete requirements. In this step, we make the questions to the users (fig. 2c), and the answers to the questions feed the FSRs patterns and, hence, create complete requirements, as shown in fig. 2(d) (e.g. *Creator* takes the value *Doctor*).

f. Group FSRs and IO attributes, by following either an object oriented approach, under Information Object, or a functional approach, under Information Object Management. Within this step, the FSRs and IO attributes are grouped, and they can take a(n): (i*) Functional orientation*: The FSRs are documented and described under one comprehensive function with the heading *IO Management*, as depicted in figure 3(a); (ii*) Object orientation:* The FSRs and IO attributes are parts of/ embedded in a particular IO, as previously illustrated in figure 3(b).

Copy Rights

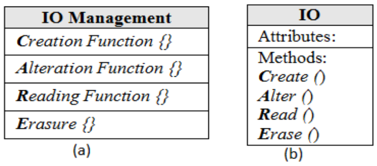


Figure 3. The methodology can take: (a) functional orientation (b) object orientation

g. *Create diagrams including Object Related Data Flow Diagrams (ORDFDs), Class Diagrams and Use Case Diagrams and Specifications.* Within this step the answers are transformed to diagrammatic notations, with the use of specific rules[4]. Figure 4(a) below shows the 2nd level ORDFD diagram for *Manage Prescription*, while figure 4(b) shows the class diagram for the *Prescription* and *Drug* IOs.

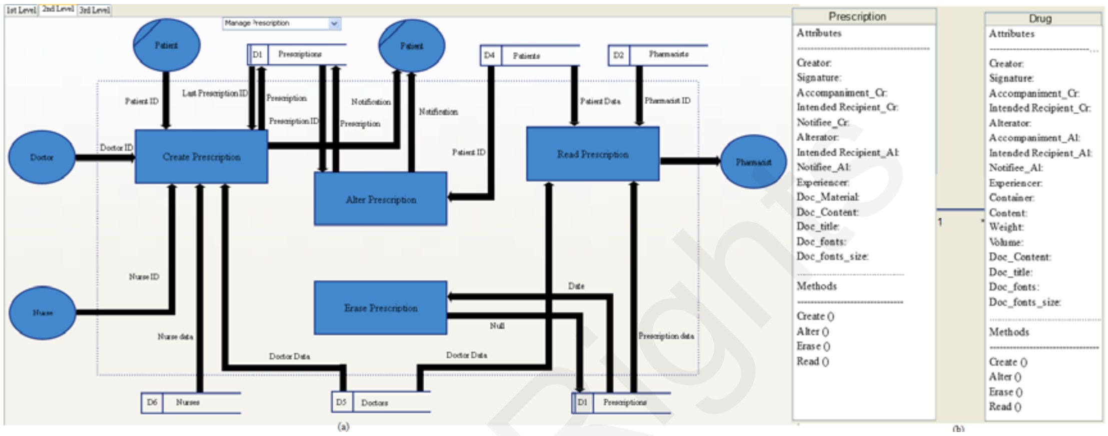


Figure 4. (a) Second Level ORDFD, (b) Class diagram

3.3 Validation and Evaluation

Requirements are considered complete when they cover functions, data, users, constraints and non-functional requirements. Our approach uses all the appropriate elements of Natural Language to cover these elements. What is further required is to elaborate in additional types of data, constraints and non-functional requirements. We have applied our methodology in a real setting which concerns the development of a new Hospital Information System, screenshot examples of which have been presented in this paper. The results have been compared to those obtained by an expert analyst who performed the same process in the same environment and have proven to be equally fast, but more accurate and exhibiting less ambiguities; at the same time it required fewer iteration with stakeholders, with meetings with stakeholders being reduced to one half compared to those conducted by the expert. More over, the resulted structured English text was easier to comprehend and agree upon on the client site (the case-study was only an experiment for research purposes and did not proceed to later stages of the Hospital IS development). Due to space limitation, we will omit further details here and this comparative evaluation will be part of a future paper. Therefore, this small-scale evaluation indicated that our methodology is efficient, reliable and provides a very strong element of validation by its nature, since the (semi) NL form of requirements are understandable to the client who gives the final approval on the requirements. To enhance understandability even more, we have also developed a parser (to be presented in a future paper) that parses the existing NL requirements and creates a purely NL Software Requirements Specification document.

Copy Rights

[4] ORDFD indicative rule: The roles of Creator, Accompaniment, Alterator, Intended Recipient, Experiencer and Notifiee correspond to actors of a traditional DFD and are represented by a circle.

4. CONCLUSION

Research studies in the area of Requirements Engineering show that there is a problem in understanding, identifying and specifying users' needs for the development of an Information System. There is lack of a methodology that provides specific steps and more importantly a formalized and understandable way to engineer requirements.

This paper has presented a methodology that is intended to formalize the major activities of Requirements Discovery, Analysis, and Specification, so that the analyst will know in advance, through a step-by-step approach, what questions to ask, in what specific way to analyse the answers to the questions, and how to write them in a specific way. The formalization is achieved with the use of natural language elements, such as verbs, nouns, genitive case, adjectives and adverbs. The key-point of the NLSSRE methodology is that Requirements Analysis and Requirements Specification use predefined types of functions and data, as well as patterns of formalized sentences, and they guide the process of building the question-sets for the Requirements Discovery stage; and the answers to these questions create complete requirements by feeding the relevant types and patterns. The proposed methodology can take an object-oriented or a functional direction. It is supported by a software tool, and with the use of specific rules, it offers automatic transformation of the resulting NL formalized requirements into diagrammatical representations, such as ORDFDs, UML class diagrams and use case diagrams, as well as use case specifications and the SRS document.

Future research steps will involve (i) application of the methodology in more real-world cases to test it further and prove its accuracy, (ii) utilization of more linguistic elements of the methodology, such as adjectives and adverbs, to form sub-IOs and sub-functions, which will be mainly used for the creation of more detailed ORDFDs and Class diagrams, as well as for implementing constraints, (iii) identification and formalisation of constraints based on the existing formalization of data and functions, (iv) identification of IOs from other objects and data, and (v) investigation of the potential of producing automatically Z specifications from the different FSR types presented in this paper.

Copy Rights

REFERENCES

Conger, S. 1994. *The New Software Engineering*, Wadsworth Publishing Company, Belmont, CA.

Fabbrini, F. et al, 2001. An Automatic Quality Evaluation for Natural Language Requirements. *Seventh International Workshop on Requirements Engineering: Foundation for Software Quality*. Interlaken, Switzerland.

Georgiades et al., 2005. A Requirements Engineering *Methodology* Based On *Natural Language* Syntax and Semantics. *13th IEEE International Requirements Engineering Conference (RE'05)*. Paris, France, pp. 73-74

Goldin, L. and Berry, D., 1997. Abstfinder: A prototype natural language text abstraction finder for use in requirement elicitation. In *Automated Software Engineering*, Vol.4, No. 4, pp. 375–412.

Guizzardi, G., 2007. Modal Aspects of Object Types and Part-Whole Relations and the de re/de dicto distinction. *19th International Conference on Advanced Information Systems Engineering (CAISE'07)*. Trondheim, Lecture Notes in Computer Science 4495, Springer-Verlag.

Li, M. et al, 2005. Weighted fuzzy interpolative reasoning method. *In Proceedings of the fourth international conference on machine learning and cybernetics.* China, pp. 3104-3108.

Rayson, P. et al, 1999. Language Engineering for the Recovery of Legacy Documents. *REVERE project report, Lancaster University*.

Rolland, C. and Proix. C., 1992. A Natural Language Approach for Requirements Engineering. *In Advanced Information Systems Engineering (P. Loucopoulos ed.),* Springer-Verlag, 257-277.

The Standish group, 2009. The CHAOS report, Press release, *http://www1.standishgroup.com/newsroom/chaos_2009.php*

Tjong, S. et al, 2006. Improving the Quality of Natural Language Requirements Specifications through Natural Language Requirements Patterns. *In Proceedings of the Sixth IEEE International Conference on Computer and Information Technology,* Seoul, Korea, pp. 199.

Videira, C., and da Silva, A., 2005. Patterns and metamodel for a natural-language-based requirements specification language. *In Proc. of the CaiSE'05 Forum*. pp. 189-194, Porto.

FEATUREOUS: INFRASTRUCTURE FOR FEATURE-CENTRIC ANALYSIS OF OBJECT-ORIENTED SOFTWARE

Andrzej Olszak and Bo Nørregaard Jørgensen
The Maersk Mc-Kinney Moller Institute
University of Southern Denmark
Campusvej 55, 5230 Odense M, Denmark

ABSTRACT

The decentralized nature of collaborations between objects in object-oriented software makes it difficult to understand how user-observable program features are implemented and how their implementations relate to each other. It is worthwhile to improve this situation, since feature-centric program understanding and modification are essential during software evolution and maintenance. In this paper, we present an infrastructure built on top of the NetBeans IDE called Featureous that allows for rapid construction of tools for feature-centric analysis of object-oriented software. Our infrastructure encompasses a lightweight feature location mechanism, a number of analytical views and an API allowing for addition of third-party extensions. To form a common conceptual framework for future feature-centric extensions, we propose to structure feature centric analysis along three dimensions: perspective, abstraction and granularity. We demonstrate feasibility of our approach by conducting a case study of change adoption in JHotDraw SVG.

KEYWORDS

Features, feature-centric analysis

Copy Rights

1. INTRODUCTION

Feature-centric analysis (Greevy, 2007) helps developers to perceive object-oriented software in terms of its user-observable behavior (Turner et al, 1999). The need for feature-centric analysis is constantly encountered during software evolution and maintenance, since users formulate their functional requirements, change requests and error reports in terms of features (Turner et al, 1999)(Mehta and Heineman, 2002). The ability to relate these descriptions to relevant fragments of object-oriented source code is a prerequisite to feature-wise modification (Greevy et al, 2007), error correction (Cornelissen et al, 2009)(Röthlisberger et al, 2007), change impact assessment (Ryder and Tip, 2001) and derivation of new features from the existing ones.

Relating features to their implementations is, however, a difficult task, since object-oriented programming languages provide no means for representing features explicitly. In object-oriented programs, features are implemented as inter-class collaborations crosscutting multiple classes as well as multiple architectural units (Murphy et al, 2001). This physical tangling and scattering of features over several units of code makes their implementations difficult to identify and understand (Turner et al, 1999)(Shaft and Vessey, 2006).

The complexity and size of feature-code mappings creates a need for a tool-supported analysis approaches. The role of tools is to guide the analysis process in a systematic fashion. Secondly, tool support is needed to automate repetitive and error-prone calculations, and thereby to ensure reproducibility and scalability of analytical activities. Finally, we deem it necessary to integrate tools for feature-centric analysis with contemporary software development environments, so that feature-centric analysis can be assimilated as part of standard activities during software evolution and maintenance.

In this paper, we aim at providing a novel tool support for feature-centric analysis of object-oriented programs. We do this by presenting our tool infrastructure for the NetBeans Java IDE (NetBeans IDE, *http://netbeans.org*) called Featureous. The infrastructure provides a lightweight dynamic feature location mechanism and an API to the basic building blocks for implementing feature-centric analytical views. In order to impose a conceptual structuring on the possible views developed on top of our tool, we propose three-dimensional categorization of feature-centric views. Thus, each view can be represented as a point on three-dimensional space of: perspectives, abstractions and granularity. Featureous is available as open-source and can be obtained from our website (Featureous tool, *http://ecosoc.sdu.dk/coe/Featureous*). This allows for immediate usage of the provided feature-centric views, building upon our infrastructure and replication of the analytical procedures presented in this paper.

In order to demonstrate feasibility of Featureous, we have used the described infrastructure for implementing a number of state-of-the-art feature-centric views. We show how these views can be applied in practice to gain insights into unfamiliar codebase of a mid-sized program. For this purpose, we analyze a subset of features of the JHotDraw project (JHotDraw framework, *http://jhotdraw.org*).

The remainder of this paper is organized as follows. In Section 2, we present the state of the art on which we base our approach. In Section 3, we give a high-level overview of our unified approach to feature-centric analysis. Section 4 describes the design of Featureous. In Section 5, we discuss the elements of our approach through their application in the JHotDraw case study. Finally, Section 6 concludes the paper.

2. STATE OF THE ART

Feature-centric analysis supports the understanding of object-oriented software by considering features as first-class analysis entities (Greevy, 2007). One of the basic elements of feature-centric analysis is the bi-directional traceability links between features and object-oriented source code. Tools that explicitly visualize this correspondence were shown to simplify discovering classes implementing a given feature and features implemented by a given class (Röthlisberger et al, 2007)(Kästner et al, 2008)(Robillard and Murphy, 2002).

By analyzing the established traceability links, it is possible to characterize features in terms of classes and characterize classes in terms of program features (Greevy and Ducasse, 2005). These characterizations can be used to investigate inter-feature relations in terms of implementation overlap. Furthermore, the static characterization based on classes can be complemented by views based on usage of objects by executing features (Salah and Mancoridis, 2004). This allows for examining run-time inter-feature dependencies.

The information contained in feature-code traceability links can be summarized by usage of software metrics. The approaches described in (Brcina and Riebisch, 2008)(Wong et al, 2000) have recognized applicability of the metrics traditionally associated with the separation of concerns to analyzing features. The two metrics proposed in (Brcina and Riebisch, 2008) - scattering and tangling - assess quantitatively the complexity of the relationships between features and computational units.

Finally yet importantly, feature location procedures are used by feature-centric analysis approaches for identification of source code fragments that contribute to implementations of program features (Wilde and Scully, 1995). The two major types of existing approaches based on static analysis (Chen and Rajlich, 2000), and dynamic analysis (Wilde and Scully, 1995)(Eisenberg and De Volder, 2005)(Olszak and Jørgensen, 2009) differ with respect automation, accuracy, and repeatability. The location approach that we adopt in this paper is a dynamic, semi-automated technique defined in (Olszak and Jørgensen, 2009). Since it relies on tracing of a program's execution, it allows for resolving polymorphic invocations, detecting common usages of objects among multiple executing features, and takes into account the effect of branch instructions on control flow.

Summing up, the existing approaches define a set of diverse methods for feature-centric analysis. Nevertheless, there is no common conceptual and technical basis for integrating them and exploring their mutual advantages. Moreover, for some of the mentioned approaches there remain questions about scalability and availability of tools implementing them. This gap we aim to fill through our approach.

Copy Rights

3. FEATURE-CENTRIC ANALYSIS OF LEGACY SOFTWARE

Feature-centric analysis is the process of analyzing programs by considering features as first-class analysis entities. What distinguishes features from the other types of source code concerns is their inherent rooting in the problem domain of programs. As the structure of object-oriented software rarely modularizes and represents features explicitly, any change tasks related to program functionality are likely to crosscut multiple units of source code and a modification to one features is likely to affect the correctness of another one that also use the fragment of code being modified.

Feature-centric analysis can be thought of as a special instance of a more general problem of *cross-decomposition analysis*. Programs can be decomposed according to various criteria and thus made to modularize different dimensions of concerns in source code. However, most of the modern programming language allow for modularizing only one dimension of concerns at a time, called the *dominant dimension* (Tarr et al, 1999). The correspondence between the modules of the dominant decomposition of a program and one of its alternative decompositions is in general of type many-to-many. This is due to the phenomena of *scattering*, where a single module of the alternative decomposition is dislocated over a number of modules of the dominant decomposition, and *tangling*, where multiple modules of the alternative decomposition are interwoven in a single module of the dominant decomposition. This lack of isomorphic correspondence between the dominant and the alternative decompositions determines changeability of programs, since different kinds of changes require different units of change. If the required change unit is modularized in the dominant decomposition, then the change can be performed in a localized manner. However, if the change unit is scattered over multiple modules, each of them will have to be modified to implement the change. It may also happen that change made to one of such under-represented concerns will result in an unforeseen modification of another one due to their tangling in terms of the same computational unit.

The mentioned situations occur for object-oriented legacy programs, if tried to be perceived in terms of feature-oriented decomposition criteria. Example of such a situation is shown in Figure 1.

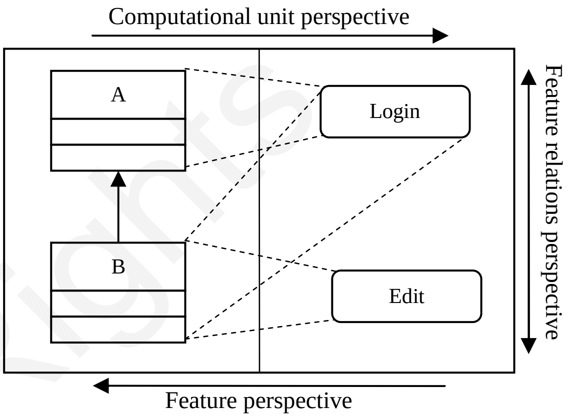


Figure 1. Perspectives on feature-code traceability links (based on (Greevy and Ducasse, 2005)).

The correspondences between object-oriented and feature-oriented decompositions of software can be investigated from four *perspectives*, based on the concrete needs of a programmer. For instance, a programmer who is given a report about an error in a particular feature would be interested in inspecting the classes that implement this feature, hence she would use the feature perspective. After the error is corrected the programmer could use the computational perspective to reason if her modifications will affect the correctness of any other features in the program. The feature relations perspective can be used by programmers to assess the overlap of implementations of two features. In summary, the three perspectives are defined as follows:

1. *Computational unit perspective* shows how computational units like packages and classes participate in implementing features (Greevy and Ducasse, 2005).
2. *Feature perspective* focuses on how features are implemented. In particular, it describes features in terms of their usage of a program's computational units (Greevy and Ducasse, 2005).
3. *Feature relations perspective* focuses on inter-feature relations that can be deduced from the feature-code mapping (Greevy, 2007).

We reckon that one of the major benefits of separating the analytical concerns by means of multiple perspectives, apart from imposing a structure on the analysis process, is reducing the complexity of analysis. This is because having multiple perspectives on the many-to-many correspondence between features and computational units allows us to avoid investigating this complex mapping directly. Instead, analysis is conducted on a number of one-to-many mappings, which are considerably easier to understand.

Within our framework, the perspectives are one of the *three dimensions* used for categorizing feature-centric analytical views. The other two dimensions are *abstraction* and *granularity*, as visualized in Figure 2.

The purpose of providing stratified levels of abstraction is to focus the analysis process by limiting the amount of information simultaneously presented to the analyst. Furthermore, stratified abstraction levels allow the complexity of a program's features to be investigated in an incremental fashion. We define three levels of abstraction:

1. *Characterization level* shows high-level diagrams, which aggregate and summarize the overall complexity of feature-code mappings.
2. *Correlation level* provides correlations between individual features and computational units.
3. *Traceability level* provides navigable traceability links between features and source code.

Copyrights

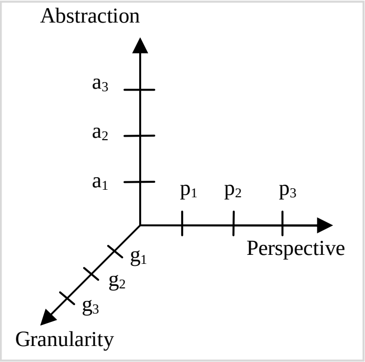


Figure 2. Three-dimensional conceptual framework.

In order to support analysis of the correspondences between features and different granularities of computational units in Java programs, we envision supporting three levels of granularity: 1. *Package granularity*; 2. *Class granularity*; 3. *Method granularity*.

Using the presented three-dimensional conceptual framework is it possible to characterize feature-centric analytical views in terms of three coordinates. For instance, view {p_2, a_1, g_2} could be a feature-class characterization in form of a plot, whereas view {p_1, a_2, g_1} could provide a correlation view of features and packages in a program in form of a graph or a table. Further examples of possible views, which have been implemented in Featureous are discussed in Section 5. Summing up, the presented conceptual framework for feature-centric analysis defines a common categorization scheme for future views implemented on top of Featureous. Thus, it structures the approaches to feature-centric analysis and allows for relating them to each other.

4. DESIGNING FEATUREOUS

The infrastructure provided by Featureous is designed around two parts: feature location mechanism and an API that exposes the trace data to feature-centric views. Featureous is implemented on top of the NetBeans Rich-Client Platform (RCP) and tightly integrated with Java IDE capabilities of the platform. The usage of the module system of NetBeans RCP allowed us to achieve extensibility of Featureous concerning adding new views by third parties without the need for recompiling the infrastructure itself.

Feature-centric analysis operates on the traceability links between features and object-oriented source code. For establishing this traceability, our approach relies on the feature location mechanism defined in (Olszak and Jørgensen, 2009). This approach requires annotating *feature entry points* in the source code of an investigated program. Feature entry points are the methods through which the execution flow enters the implementations of features. In case of GUI programs, in which features are triggered through GUI elements, feature entry points will most often be the *actionPerformed* methods of event-handling anonymous classes. Feature entry point annotations placed on method declarations have to be parameterized by the string-based identifiers of their corresponding features. Based on the annotations inserted by a programmer in the code, it is possible to locate implementations of features by tracing the execution of the program when a user is interacting with it. To achieve this, the program is instrumented with a tracing agent that registers information about the execution in the control context of feature entry points. The tracing process is transparent for the program user and it does not introduce a significant performance and memory overhead, since it does not register the information about the timing and order of captured events. Featureous integrates the feature location process with NetBeans IDE by providing a new execution button in the IDE, which transparently instruments the program before executing it.

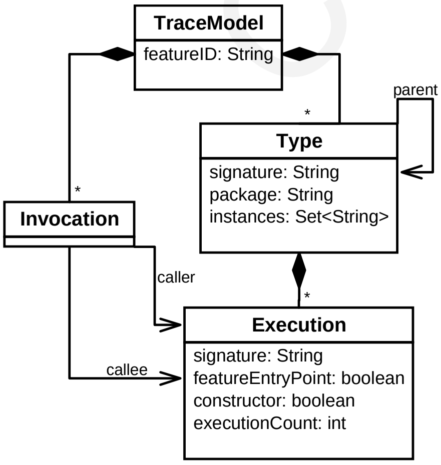


Figure 3. Feature trace model of Featureous.

The feature location process produces a set of feature traces that contain a mapping between features and source code of the program. The model that we use to represent a trace of a single feature is shown in Figure 3. Feature trace models, being an input the feature-centric analysis, contain the information about packages, methods, constructors, classes, instances, and inter-method invocations that occurred at run-time in implementations of features. This data is then exposed through an API to feature-centric views.

The usage of the API and implementing of an example feature-centric view is demonstrated in Figure 4.

The access to feature trace models is obtained through the *Controller* singleton class contained in the core module of Featureous. Apart from providing the access to trace set, *Controller* exposes a number of helper methods, such as: loading and unloading of traces, splitting and merging traces, defining the global relation of similarity between traces (e.g. based on code sharing vs. based on instances co-usage) and defining the *affinity categories* (Greevy and Ducasse, 2005). The affinity categories can be used by the views to enrich their representations and to provide a correspondence to other views. Depending on the level of participation in implementing features, the three affinity categories determine whether a computational unit is an infrastructural unit (used by at least 50% of features), a group-feature unit (used by more than one, but less than 50% of features), or a single-feature unit. The affinities are represented by their respective colors: green, blue, and red. We enhance the original representation of affinities to display the shades of affinity color hinting how strongly a computational unit belongs in an affinity category. The darker the color the more features a computational unit belongs to. The affinity coloring indicates the level of reuse of computational units across features.

The example view implemented in Figure 4 is created by extending the *GenericTraceView* abstract class and implementing its abstract methods. The three abstract methods are used in template method pattern in the base class and are called by it upon the creation, update of trace data and closure of the view. A view class implemented in the presented way is dynamically discovered by Featureous without the need for their presence at compile time. Each of the found views is then represented by a button in the main toolbar of Featureous. Automatic discovery of views is done by using the lookup mechanism of NetBeans RCP. Featureous looks up all the providers of the *FeatureTraceView* service, and adds them to the toolbar. The example view in Figure 4 is declared as a service provider by annotating the declaration of its class.

```
@ServiceProvider(service=FeatureTraceView.class,position=7)
public class ExampleView extends GenericTraceView {

  public ExampleView() {
    super("ExampleView", null, ExampleView.class);
    setName("This is an example view");
  }

  public void createView() {
    Controller c = Controller.getInstance();
    Set<TraceModel> ftms = c.getTraceSet().getAllTraces();
    String msg = "No. of traces loaded: " + ftms.size();
    JLabel status = new JLabel(msg);
    this.setLayout(new BorderLayout());
    this.add(status, BorderLayout.CENTER);
  }

  public void updateView() { ... }

  public void closeView() { ... }
}
```

Figure 4. Extending Featureous.

Based on the mechanism for extending Featureous with new views and the API that exposes feature trace models, we have implemented a number of feature-centric views defined in the literature. During this process, we have discovered that some of the views are complementary to each other and can be combined and that all the views can be adapted to use the global affinity-coloring scheme discussed earlier.

Copy Right

5. FEATURE-CENTRIC MODIFICATION – A CASE STUDY

In this section, we use a case study to present three feature-centric views and demonstrate their application to supporting a feature-centric modification of a legacy object-oriented program.

The case study being presented is concerned with a program built on top of the JHotDraw 7.2 framework called SVG (JHotDraw framework, *http://jhotdraw.org*). SVG is a vector graphic-based drawing editor for Java. The program consists of 62K lines of code and contains significantly high number of features for the case study to be considered a realistic application scenario. The task under investigation was to modify the *export* feature of SVG so that a *watermark text* is added to any exported drawing file. It is worth mentioning that prior to conducting this case study we had no significant exposure to the implementation details of SVG.

First, we had to establish traceability links between features and source code of SVG. To achieve this, we needed to recover the list of features of SVG, since no requirement specification documents were available. In order to identify features of SVG, we have inspected the executing application. We have performed this by investigating user-triggerable functionality in graphical user interface elements like the main menu, contextual menus, and toolbars. We have identified 28 features, whose 91 feature entry point methods we have annotated in JHotDraw's source code. By manually triggering each identified feature at run-time in the instrumented SVG program, we obtained a set of feature traces that were the input to feature-centric analysis.

Firstly, we wanted to estimate the effort needed to perform the intended modification task. This was done using our enhanced version of the feature characterization view (Greevy and Ducasse, 2005). This view can be specified in terms of our conceptual analysis framework presented in Section 3 as {p_2, a_1, $g_{1,2}$}. Feature characterization view is designed as a bar chart summarizing implementations of features. Each feature is represented here by a separate bar, by whose height we indicate the scattering (Brcina and Riebisch, 2008) of feature over computational units (either packages or classes). The coloring of bars shows the distribution of the computational units within the affinity-based categories. In addition, this information is shown within bars also as a distribution profile plot. This fine-grained information on the characterization of contributing computational units gives an impression on how difficult it would be to change the implementation of a given feature without affecting the rest of a program's functionality. I.e. changing a red-colored unit will only affect the feature itself, whereas changing green or blue units will affect other features as well.

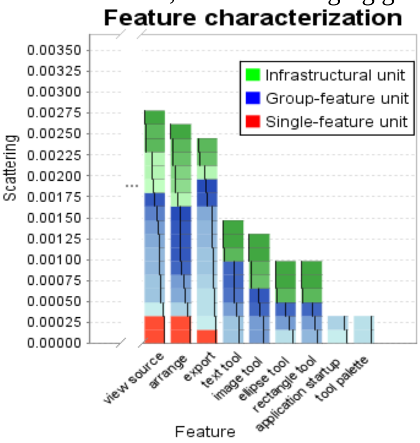


Figure 5. Feature characterization of SVG.

The results of feature characterization obtained for SVG for granularity of classes are shown in Figure 5. It can be seen that the *export* feature, which is responsible for exporting drawings from the program's canvas to various file formats, contains only one feature-specific class. This indicates that there exists a high chance that our modification of this feature will affect the correctness of some other features in the program due to the high degree of code sharing. Another feature that we are interested in, in the context of our modification task, is the *text tool* feature. Because this feature is responsible for drawing the text-based shapes on the program canvas, it should contain classes that we have to use to programmatically draw a watermark text in exported drawings. The relatively low value of scattering of this feature indicates that we will not have to visit many classes in order to find the ones that we need to reuse.

In order to identify the classes that are used by *text tool* for creating a text figure on canvas we used the navigable trace inspector. Trace inspector provides traceability from features to contents of feature traces and is defined in terms of our conceptual framework as {p_2, a_3, $g_{1,2,3}$}. Trace inspector's window, depicted in Figure 6, contains a hierarchy of nodes symbolizing the packages, classes and methods that implement a feature. The tree nodes symbolizing classes and methods can be used for automatic navigation to their corresponding source code fragments in the NetBeans editor. The methods annotated as feature entry points are marked in the hierarchy tree using a distinct icon, due to their special role in implementations of features.

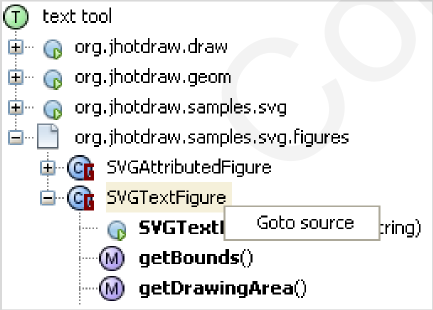

text tool
org.jhotdraw.draw
org.jhotdraw.geom
org.jhotdraw.samples.svg
org.jhotdraw.samples.svg.figures
SVGAttributedFigure
SVGTextFigure
SVGTextI Goto source :ring)
getBounds()
getDrawingArea()

Figure 6. Navigable trace inspector.

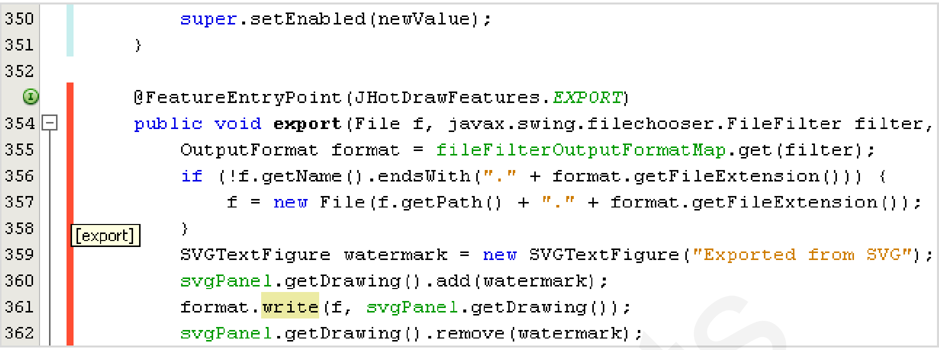
By visiting the classes that contribute the text tool feature and using the navigable traceability to their source codes, we have located the class *SVGTextFigure*, which is the candidate class that we will use for creating watermarks in exported drawings.

We use the same view to find the class that performs the export of drawings. As the export activity consists of a chain of invocations that involves many classes, there exist many possible places where a drawing could be equipped with a watermark before exporting. We choose to do it in the *SVGView* class that implements a view for a single SVG drawing. This class contains an export method that we can modify to achieve our goal and it aggregates a drawing panel, through which it is possible to access and modify the drawing being exported. Usage of feature-centric analysis was of significant help during localization of this class, because of ubiquitous usage of polymorphism in the JHotDraw framework. The indirection provided by polymorphism would otherwise be a significant obstacle to finding the concrete classes that carry out export-related functionality in SVG.

To add the watermark to a drawing being exported we have modified the *export* method of the *SVGView* class. In order to ensure that the modifications made will have no impact on other features of the program (e.g. the *drawing persistence* feature, which could also invoke the *export* method), we have used the editor coloring view of Featureous. Featureous enhances the default code editor of the NetBeans IDE to provide feature-centric information about participation of source code in implementations of features. This view can

Copyright

be defined in terms of our conceptual framework as {p_1, a_3, $g_{2,3}$}. Figure 7 shows how Featureous uses color bars next to the editor to visualize the affinity of the viewed source code fragments. The bars allow for immediate assessment of source code with respect to its participation in features. Furthermore, traceability from source code to concrete features is provided in form of tooltips associated with the color bars. The functionality provided by the editor coloring not only supports the understanding of source code in terms of features it implements, but also simplifies the reasoning about possible consequences of source code modifications on the correctness of program's functionality.



```
350         super.setEnabled(newValue);
351     }
352
        @FeatureEntryPoint(JHotDrawFeatures.EXPORT)
354     public void export(File f, javax.swing.filechooser.FileFilter filter,
355         OutputFormat format = fileFilterOutputFormatMap.get(filter);
356         if (!f.getName().endsWith("." + format.getFileExtension())) {
357             f = new File(f.getPath() + "." + format.getFileExtension());
358         }
359         SVGTextFigure watermark = new SVGTextFigure("Exported from SVG");
360         svgPanel.getDrawing().add(watermark);
361         format.write(f, svgPanel.getDrawing());
362         svgPanel.getDrawing().remove(watermark);
```

Figure 7. The performed modification in colored code editor.

As shown in Figure 7, we add a simple watermark to the drawing (lines 359, 360) before it is written to the output file by invoking *format.write()* and remove it afterwards (line 362), so that the drawing being further edited in the SVG program is not altered with the watermark. During the implementation of this change, we used the affinity-colored bars to ensure that our modification will not affect features other than *export*. As indicated by editor coloring, the *export* method is feature-specific, even though the enclosing class participates in four other features.

Summary of the case study. In this case study, we have modified one of the features of the JHotDraw SVG program. The change adoption process was supported by three feature-centric views built on top of the Featureous infrastructure. It is our experience that the usage of feature-centric analysis reduced the extent of necessary investigations of unfamiliar source code and allowed us to reason about the impact of the performed modification for the overall correctness of the program.

Copy Rights

6. CONCLUSION

As mentioned previously, there exists a lack of isomorphic correspondence between the users' and the programmers' perception of object-oriented programs. This becomes problematic during software evolution and maintenance, because the implementations of features requested by the users to be modified are not evident from the object-oriented source code. Hence, there is an urgent need for tool-supported analysis approaches, which will help developers to understand the correspondence between features and code.

In this paper, we have presented our solution: a novel tool infrastructure and conceptual framework for implementing feature-centric analytical views of legacy object-oriented programs. Our infrastructure called Featureous is implemented as a plug-in to the NetBeans IDE. Featureous provides a lightweight mechanism for recovering the feature-code traceability links and an API for implementing third-party extensions. Our tool infrastructure comes with implementations of a number of state-of-the-art feature-centric views, three of which we have presented in our case study. The case study performed on JHotDraw SVG demonstrates how feature-centric analysis can be used to aid modification tasks during software evolution and maintenance.

We hope that the provided infrastructure Featureous will help researchers to experiment with feature-centric analysis of software. Motivated by our experiences reported in this paper, we believe that usage of feature-centric analysis tools such as Featureous can improve the performance of adopting functionality-related changes during software evolution. In a long perspective, we hope that improved understanding of feature-code relations may improve the practices of implementing features in object-oriented programs.

ACKNOWLEDGEMENT

We would like to thank Kemal Pajevic for providing feature analysis software – a predecessor of Featureous.

REFERENCES

Brcina, R. and Riebisch, M., 2008. Architecting for evolvability by means of traceability and features. Automated Software Engineering - Workshops. ASE Workshops 2008. 23rd IEEE/ACM International Conference on. pp. 72-81.

Chen, K. and Rajlich, V., 2000. Case Study of Feature Location Using Dependence Graph. IWPC '00: Proceedings of the 8th International Workshop on Program Comprehension. p. 241 IEEE Computer Society, Washington, DC, USA.

Cornelissen, B. et al, 2009. Trace Visualization for Program Comprehension: A Controlled Experiment. In: Marcus, A. and Koschke, R. (eds.) Proceedings of the 17th International Conference on Program Comprehension (ICPC'09). pp. 100–109 IEEE Computer Society, Washington, DC, USA.

Eisenberg, A.D. and De Volder, K., 2005. Dynamic Feature Traces: Finding Features in Unfamiliar Code. ICSM '05: Proceedings of the 21st IEEE International Conference on Software Maintenance. pp. 337–346 IEEE Computer Society, Washington, DC, USA.

Greevy, O. and Ducasse, S., 2005. Correlating Features and Code Using a Compact Two-Sided Trace Analysis Approach. CSMR '05: Proceedings of the Ninth European Conference on Software Maintenance and Reengineering. pp. 314–323 IEEE Computer Society, Washington, DC, USA.

Greevy, O. et al, 2007. How Developers Develop Features. CSMR '07: Proceedings of the 11th European Conference on Software Maintenance and Reengineering. pp. 265–274 IEEE Computer Society, Washington, DC, USA.

Greevy, O., 2007. Enriching Reverse Engineering with Feature Analysis. PhD thesis. University of Bern.

Kästner, C. et al, 2008. Granularity in Software Product Lines. Proceedings of the 30th International Conference on Software Engineering (ICSE). pp. 311–320 ACM, New York, NY, USA.

Mehta, A. and Heineman, G.T., 2002. Evolving legacy system features into fine-grained components. ICSE '02: Proceedings of the 24th International Conference on Software Engineering. pp. 417–427 ACM, New York, USA.

Murphy, G.C. et al, 2001. Separating Features in Source Code: An Exploratory Study. ICSE 2001: International Conference on Software Engineering, , p. 0275.

Olszak, A. and Jørgensen, B.N., 2009. Remodularizing Java programs for comprehension of features. Proceedings of International Workshop on Feature-Oriented Software Development. pp. 19–26 ACM, New York, USA.

Robillard, M.P. and Murphy, G.C., 2002. Concern graphs: finding and describing concerns using structural program dependencies. ICSE '02: Proceedings of the 24th International Conference on Software Engineering. pp. 406–416 ACM, New York, NY, USA.

Röthlisberger, D. et al, 2007. Feature driven browsing. ICDL '07: Proceedings of the 2007 international conference on Dynamic languages. pp. 79–100 ACM, New York, NY, USA.

Ryder, B.G. and Tip, F., 2001. Change impact analysis for object-oriented programs. Proceedings of the 2001 ACM SIGPLAN-SIGSOFT workshop on Program analysis for software tools and engineering. pp. 46–53 ACM, New York, USA.

Salah, M. and Mancoridis, S., 2004. A Hierarchy of Dynamic Software Views: From Object-Interactions to Feature-Interactions. ICSM '04: Proceedings of the 20th IEEE International Conference on Software Maintenance. pp. 72–81 IEEE Computer Society, Washington, DC, USA.

Shaft, T. and Vessey, I., 2006. The Role of Cognitive Fit in the Relationship Between Software Comprehension and Modification. MIS Quarterly, 30(1). pp. 29-55.

Tarr, P. et al, 1999. N degrees of separation: multi-dimensional separation of concerns. ICSE '99: Proceedings of the 21st international conference on Software engineering. New York, NY, USA: ACM, pp. 107-119.

Turner, C.R. et al, 1999. A conceptual basis for feature engineering. In *Journal of Systems and Soft.*, vol. 49, pp. 3–15.

Wilde, N. and Scully, M.C. 1995. Software reconnaissance: mapping program features to code, Journal of Software Maintenance, vol. 7, pp. 49–62.

Wong, W.E. et al, 2000. Quantifying the closeness between program components and features, J. Syst. Softw., vol. 54, pp. 87–98.

Copy Rights

STRUCTURED AND FLEXIBLE GRAY-BOX COMPOSITION: APPLICATION TO TASK RESCHEDULING FOR GRID BENCHMARKING

Ismael Mejía and Mario Südholt
ASCOLA team (EMN-INRIA, LINA), Département Informatique, École des Mines de Nantes, France

ABSTRACT

The evolution of complex distributed software systems often requires intricate composition operations in order to adapt or add functionalities, react to unanticipated changes to security policies, or do performance improvements, which cannot be modularized in terms of existing services or components. They often need controlled access to selected parts of the implementation, *e.g.*, to manage exceptional situations and crosscutting within services and their compositions. However, existing composition techniques typically support only interface-level (black-box) composition or arbitrary access to the implementation (gray-box or white-box composition).

In this paper, we present a more structured approach to the composition of complex software systems that require invasive accesses. Concretely, we provide two contributions, we *(i)* present a small *kernel composition language for structured gray-box composition* with explicit control mechanisms and a corresponding aspect-based implementation; *(ii)* present and compare evolutions using this approach to gray-box composition in the context of two real-world software systems: benchmarking of grid algorithms with NASGrid and transactional replication with JBoss Cache.

KEYWORDS

Software Composition, Software Engineering, Distributed Software

Copy Rights

1. INTRODUCTION

The evolution of large-scale distributed software systems often requires the unanticipated introduction of new functionalities or the modification of existing ones. Such evolution tasks are often inherently difficult because of two fundamental problems. First, the compositions cannot be expressed only in terms of the interfaces of the involved systems (non-invasive modifications) but also imply changes to some (typically limited) parts of the corresponding implementations. Second, the compositions often involve functionalities that are not well modularized in the existing systems or in the resulting composed system. Such composition problems occur frequently in legacy ERP systems that, *e.g.*, to cope with new security requirements imposed by changing legal frameworks, such as the Sarbanes-Oxley act in the U.S. (such evolution problems for SAP AG's SOA infrastructure are considered, *e.g.* in the CESSA[1] research project).

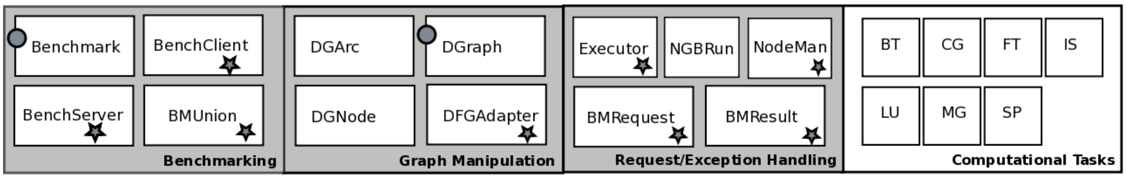


Figure 1. NASGrid: application structure and scheduling-relevant code parts

As a concrete real-world example (that we will consider in more detail later on), we have studied NASA's NASGrid benchmarking infrastructure for computational grids (Frumkin R. et al, 2001). This benchmark is used to time grid computations that may execute on different communication topologies. Fig. 1 shows the

[1] This work has been supported by the CESSA project (ANR 09-SEGI-002-01, see cessa.inria.gforge.fr).

main components, shown with gray background, of the NASGrid benchmarking frameworks: three sets of classes that respectively provide a benchmarking interface, exception handling (principally of network conditions), and management of the graph structure representing the graph communication topology (the remainder of the system being constituted essentially by routines for numerical algorithms, called computational tasks in the figure).

NASGrid basically executes computations on distributed nodes and forwards intermediate results according to communication dependencies defined in terms of a topology graph. Grid computations are aborted in the case of exceptions, such as severe network errors; task rescheduling in the case of exceptions is not supported. We have investigated an evolution of NASGrid to add this useful functionality that fits well with existing, frequently long-running, grid applications. Our analysis of the existing code base has shown that the extension of NASGrid by task rescheduling partially requires modifications to the existing interfaces (*i.e.*, sets of public classes and methods that are marked by disks in Fig. 1). However, the extension also requires some access to the NASGrid implementation because the necessary modifications as a whole are crosscutting with respect to the existing structure of NASGrid (the corresponding classes are marked by stars in Fig. 1).

Performing such evolutions using mainstream languages or development methods is highly difficult and error prone: (i) the crosscutting nature of such evolutions involve a potentially large number of modifications that have to be carefully synchronized; (ii) structural means and semantic properties should be supported in order to control the effects of invasive modifications to implementations.

In this paper we present an approach of structured invasive, *i.e.*, gray-box, composition that supports accesses to interfaces and implementations through compositions of basic programming patterns for invasive access, resulting in gray-box compositions whose degree of invasiveness and their impact on an implementation can be controlled explicitly and flexibly. Furthermore, these operators allow crosscutting functionalities that are part of the subsystems to be expressed modularly.

Concretely, we present two contributions: First, we introduce in section 2 a kernel language for invasive composition that enables explicit and expressive compositions of invasive distributed patterns (Benavides L. et al, 2007) (henceforth simply called invasive patterns). Invasive patterns and compositions thereof provide flexible control of gray-box compositions and support the modularization of crosscutting functionalities using aspect-oriented programming techniques (Kiczales G, 1996). We also briefly present an implementation of this kernel language using the AWED system (Benavides L. et al, 2006), (AWED website, 2010) for explicitly-distributed AOP. Second, in section 3 we present and evaluate how our approach supports an evolution scenario of NASGrid that add task rescheduling. This extension is non-trivial, interacting with the original application at 28 places and is modularly implemented by an aspect and four new ordinary classes. Finally, we briefly compare the corresponding composition properties with those of two other case studies we have performed as part of previous work: a less invasive evolution of NASGrid for checkpoint introduction; and a highly invasive evolution of JBoss Cache, a middleware for transactional replication of data in distributed systems.

Our results show that invasive compositions allow a whole space of evolutions that require invasive modifications to be expressed while maintaining much higher control of the impact of invasive modifications, and this for systems requiring from moderately invasive to highly invasive accesses. As to our knowledge, no other approach to gray-box composition has been applied to such a range of evolution scenarios nor provides a comparable level of control of effects.

2. STRUCTURED AND FLEXIBLE INVASIVE COMPOSITION

Evolution scenarios as discussed in the previous section require three essential requirements to be addressed:

R1) Enable (modifications to) the coordination of distributed communications and computations.

R2) Support modularization of crosscutting functionalities that are subject to evolution tasks.

R3) Provide structural and property-based control over modifications, in particular invasive ones.

From a general point of view, we address these three issues as follows: we exploit invasive patterns as basic abstractions in order to express coordination and communication requirements of distributed applications that involve crosscutting functionalities. We introduce a composition language over patterns that enables the definition of structured and flexible pattern compositions whose effects may be controlled, *e.g.*, by limiting invasive accesses to contexts defined by event sequences. In the remainder of this section we briefly revisit the notion of invasive patterns for distributed programming and then define a kernel language for the flexible composition of such patterns. Finally, we show how such pattern compositions can be implemented in terms of the AWED system (Benavides L. et al, 2006), a system for distributed aspects.

2.1 Invasive Patterns

Invasive patterns (Benavides L. et al, 2007) have been introduced as generalizations of standard parallel and distributed programming patterns. Fig. 2 shows the three invasive patterns we consider: a gather, a farm and a pipelining pattern. All of these patterns match sequences of execution events (illustrated by the dotted curves) over calls to interface methods or methods called in the implementation. These sequences are matched on one or several source nodes, construct data (using a computation represented by the filled rectangle on the source nodes) that is sent to a number of one or several target nodes and integrated into the computations there (as represented by the filled rectangle on the target side). Invasive patterns allow quantifying over sets of source and target nodes, in particular, the event sequences that trigger actions as well as the actions themselves.

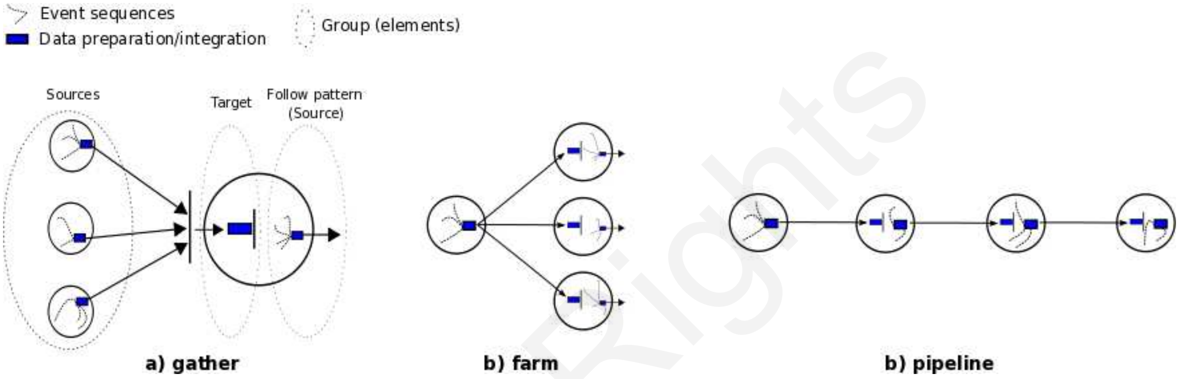


Figure 2. Invasive patterns

Invasive patterns provide basic support for the three requirements mentioned above: as frequently used patterns for distributed programming, they support distribution coordination (R1); modularization of crosscutting functionalities (R2) can be achieved by means of aspects for the definition of event sequences (history-based pointcuts in AOP-speak) and actions (advice in AOP-speak) that compute data to be transferred from source to target nodes and that integrate data into target computations. Finally, some control over accesses and computations is provided by their fixed overall structure.

Copy Rights

2.2 A Kernel Language for Non/Invasive Composition

In this paper, we introduce a composition language over invasive patterns in order to fully address the requirements for evolution tasks. We strive, in particular, for a language that enables flexible compositions of patterns to handle more complex crosscutting functionalities and provides better control over, possibly invasive, modifications performed by pattern compositions.

$$
\begin{array}{llll@{\qquad}llll}
\mathit{Prog} & ::= & \overline{\mathit{Op}} & \text{; Programs} & \mathit{Adap} & ::= & e \mid e_G \mid P \mid \overline{\mathit{Adap}} & \text{; Adaptations} \\
\mathit{Op} & ::= & (\mathit{Ctx}, \mathit{Adap}) & \text{; Operators} & P & ::= & (\overline{\mathit{Op}}, \overline{\mathit{Op}}) & \text{; Patterns} \\
 & & & & & & & \\
\mathit{Ctx} & ::= & e \mid e_G \mid \overline{\mathit{Ctx}} & \text{; Contexts} & G & ::= & \texttt{if}(B) \mid \overline{h} & \text{; Guards}
\end{array}
$$

Figure 3. Kernel language for invasive composition

Fig. 3 presents the essentials of our kernel language for invasive composition. Some remarks on notation: non-terminals, such as *Op* or *P* start with an upper case letter and are set in italic font; lexical categories, such as *e* are denoted by lower case, italic letters; terminals, such as if set in typewriter font. *X* denotes finite sequences of expressions of non-terminal *X*. (A more elaborate version of the language that supports, *e.g.*, repetitions in form of regular expressions is in preparation but not needed for the extension of the NASGrid application by dynamic task rescheduling considered in this paper).

The intuition behind this core language is as follows: operators match contexts that trigger sequences of adaptations. Contexts are built from event sequences that may be guarded. Adaptations include simple manipulations enabling the insertion of glue code, such as communication statements, but also potentially complex pattern compositions built from the three invasive patterns introduced above.

The grammar defines four main syntactic categories: (evolution) programs *Prog*, operators *Op*, contexts *Ctx* and adaptations *Adap*. *(Evolution) programs* are sequences over evolution operations (instantiations of invasive patterns or pattern compositions). An *operator* is defined as a pair of a context and an adaptation. *Contexts* consist of sequences of guarded events (cf. *G*), *i.e.*, events that may be matched on specific hosts or under specific conditions (represented by *B*, the nature of which is unspecified here; typically we expect conditions of limited expressiveness to support property analysis and verification).

Adaptations come in two forms: sequences of (i) possibly guarded events that represent (computation or communication) glue code potentially triggered on specific hosts and under specific conditions; (ii) structured adaptations in form of *pattern compositions P* that are pairs of sequences of operators. Such a pattern, say *(s,t)* denotes adaptations on sources *s* and targets *t*, typically the extraction of data on sources that are send for further handling to the targets. Note that patterns and pattern compositions may form both the context and adaptation parts of the operators.

Our language directly supports very flexible pattern compositions. As a simple example (that has been applied for NASGrid task rescheduling) consider an application of a farm-pattern followed by a gather pattern. The farm will match an event sequence on one node, extract information, send and inject it into a number of target nodes. The gather pattern will then monitor for event sequences on its source nodes that, in its simplest case, are the target nodes of the farm pattern, extract information on the source nodes of the gather pattern and inject them in its target node.

2.3 Implementation using distributed Aspects and AWED

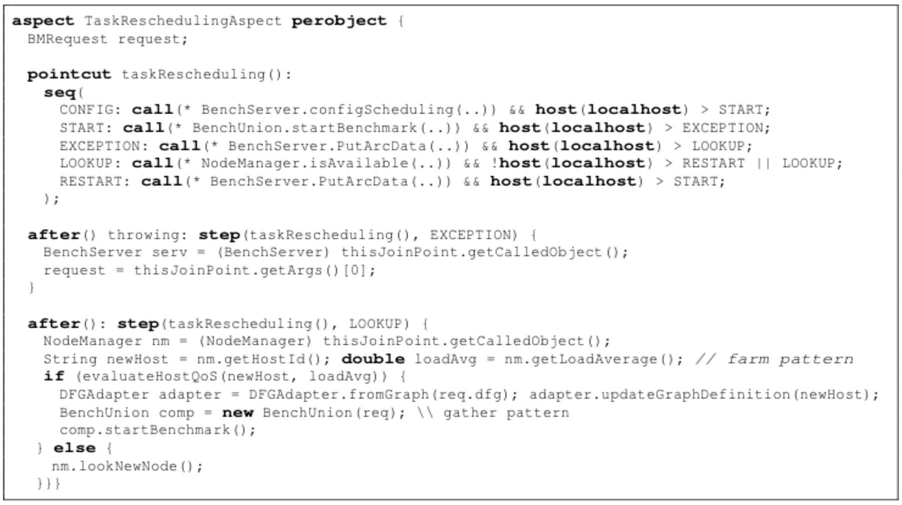
The AWED system (Benavides L. et al, 2006) provides an aspect model for distributed systems that provides means for the monitoring of sequences of events, history-based pointcuts in AOP-speak, that occur on different (groups of) hosts. Such event sequences are described in terms of guarded finite-state systems; AWED also provides various means to trigger actions, advice in AOP, where the corresponding pointcut-defining event sequences are matched.

The language above can be implemented using AWED in terms of event sequences that define the, interface level or implementation-level, context (*Ctx* in the above grammar) and use actions to define adaptations (*Adap*). Invasive patterns (farm, gather and pipeline) are then implemented as pairs of aspects corresponding to source and target computations of the patterns. Pattern compositions, say p_2 *o* p_1, are implemented by aspects that match end-marking events in p_1 and trigger execution of p_2. We have implemented task rescheduling for NASGrid using AWED this way.

Fig. 4 shows the main component of the implementation of the task rescheduling aspect. Here, the pointcut taskRescheduling (lines 4–11) defines a mixed interface/implementation-level context that identifies exception occurrences (state EXCEPTION) and, possibly repeated, choices of alternative available hosts (state LOOKUP). The second advice (lines 18–27) chooses an alternative and restarts the benchmark (*i.e.*, triggers a farm pattern that sends info to the successor nodes of the current one). Overall, this language provides flexible structured invasive access through pattern compositions that may be subjected to explicit control through predefined compositions and the precise definition of application contexts by means of event sequences.

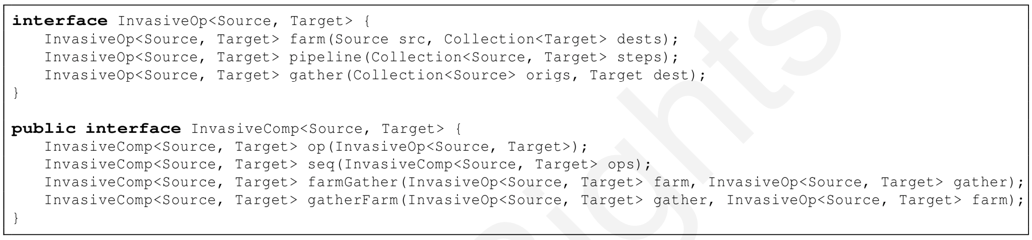
Implementing composition of invasive patterns. Fig. 5 shows an interface we have developed that represents a subset of the above language that makes explicit invasive patterns and pattern compositions. The pattern composition constructors enable building of compositions from simple operators (constructor op), sequences of compositions (seq), and compositions of farm and gather patterns (the latter two being expressible as sequences and are necessary for the task rescheduling example).

Copy Rights



```
aspect TaskReschedulingAspect perobject {
  BMRequest request;

  pointcut taskRescheduling():
    seq(
      CONFIG: call(* BenchServer.configScheduling(..)) && host(localhost) > START;
      START: call(* BenchUnion.startBenchmark(..)) && host(localhost) > EXCEPTION;
      EXCEPTION: call(* BenchServer.PutArcData(..)) && host(localhost) > LOOKUP;
      LOOKUP: call(* NodeManager.isAvailable(..)) && !host(localhost) > RESTART || LOOKUP;
      RESTART: call(* BenchServer.PutArcData(..)) && host(localhost) > START;
    );

  after() throwing: step(taskRescheduling(), EXCEPTION) {
    BenchServer serv = (BenchServer) thisJoinPoint.getCalledObject();
    request = thisJoinPoint.getArgs()[0];
  }

  after(): step(taskRescheduling(), LOOKUP) {
    NodeManager nm = (NodeManager) thisJoinPoint.getCalledObject();
    String newHost = nm.getHostId(); double loadAvg = nm.getLoadAverage(); // farm pattern
    if (evaluateHostQoS(newHost, loadAvg)) {
      DFGAdapter adapter = DFGAdapter.fromGraph(req.dfg); adapter.updateGraphDefinition(newHost);
      BenchUnion comp = new BenchUnion(req); \\ gather pattern
      comp.startBenchmark();
   } else {
     nm.lookNewNode();
   }}}
```

Figure 4. Implementation of task rescheduling in NASGrid using AWED



```
interface InvasiveOp<Source, Target> {
    InvasiveOp<Source, Target> farm(Source src, Collection<Target> dests);
    InvasiveOp<Source, Target> pipeline(Collection<Source, Target> steps);
    InvasiveOp<Source, Target> gather(Collection<Source> origs, Target dest);
}

public interface InvasiveComp<Source, Target> {
    InvasiveComp<Source, Target> op(InvasiveOp<Source, Target>);
    InvasiveComp<Source, Target> seq(InvasiveComp<Source, Target> ops);
    InvasiveComp<Source, Target> farmGather(InvasiveOp<Source, Target> farm, InvasiveOp<Source, Target> gather);
    InvasiveComp<Source, Target> gatherFarm(InvasiveOp<Source, Target> gather, InvasiveOp<Source, Target> farm);
}
```

Figure 5. Invasive Composition Interface

3. STRUCTURED AND FLEXIBLE INVASIVE COMPOSITION

In this section we consider the implementation of evolution scenarios using invasive patterns in the context of two real-world software systems of medium size, the NASGrid application (ca. 21 KLOC) and the JBoss Cache middleware for distributed caching under transactional control (ca. 50 KLOC). Concretely, we present the task rescheduling evolution for NASGrid in more detail, especially its use of invasive composition and how our language can be used to exert control over invasive modifications. Furthermore, we briefly compare the composition characteristics of the task rescheduling case study with two other evolution scenarios that we have previously performed using invasive patterns but without explicit support for the composition of patterns. This comparison shows that compositions of invasive patterns allow to cover a whole range of evolution scenarios, from limited invasive ones to highly invasive ones and that the flexible pattern compositions our language supports simplify such evolution tasks.

3.1 Invasive Composition for NASGrid Task Rescheduling

NASA's NASGrid benchmark allows to time grid applications that are deployed on different hardware topologies; communication paths taken as part of a grid application are represented in NASGrid using a topology graph. Each computation on a node is modeled using an individual worker thread that executes some numerical computation, using building blocks, such as LU matrix decomposition and Fourier transforms (FT). These computational tasks are supervised by a coordinator thread which forwards the results to other nodes as defined by the topology graph.

The main obstacle for adding task rescheduling as a strategy for improving fault tolerance in the case of network or node failures, is the static topology representation and benchmark execution in NASGrid. More concretely, in terms of the NASGrid system architecture shown in Fig. 1, the graph manipulation part does not accommodate topology changes, and the benchmarking part does not allow to probe the status of network

Copy Rights

connections, or to test the availability of remote hosts, or to modify the routing of data between nodes. Our extension to NASGrid introduces these features and exploits them when exceptional situations occur. In order to achieve that goal we have to extend the interfaces (of interfaces and classes marked disks in the diagrams) and the implementation of the classes (that are marked by stars in the figure) at multiple points. Note that a almost all disks or stars represent several modifications within the same interface or class. Overall, NASGrid has to be modified at 28 locations to extend it modularly by the task rescheduling functionality.

In order to give a concrete idea of which code manipulations are involved in invasive accesses and pattern compositions, let us first have a look at the code excerpt shown in Fig. 6. This excerpt shows the NASGrid code for localization and propagation of data between nodes. In case that a remote node is unavailable (lines 14-17) no reaction is taken and the exception is only passed along. However, as this method makes explicit the data on the real successor nodes of the current node (lines 4-8), we have to access it to update the new node information with the corresponding data after rescheduling.

```
1  public int PutArcData(BMRequest req,BMResults res)
2      throws RemoteException {
3      DGNode nd=req.dfg.node[req.pid];
4      BMRequest lreq[]=new BMRequest[nd.outDegree];
5      // ... process info
6      for(int i=0;i<nd.outDegree;i++){
7          lreq[i]=new BMRequest(req);
8          // ... clone arq info
9          try {
10             Benchmark RemBench = (Benchmark) Naming.lookup("//"+
11                     lreq[i].MachineName+"/BenchmarkServer");
12             lreq[i].tmSent=System.currentTimeMillis();
13             RemBench.SendData(lreq[i],res);
14         } catch (Exception e) {
15           // ... print exception stacktrace
16           throw new RemoteException("BenchServer exception: ", e);
17 }}}
```

Figure 6. Class BenchServer (fragment) task rescheduling

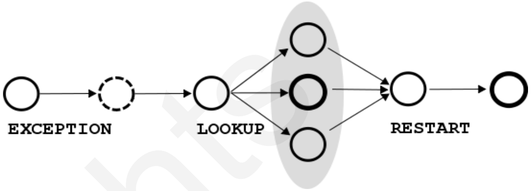


Figure 7. NASGrid invasive composition

We have extended NASGrid with the help of invasive composition operators implemented with sequences in AWED as presented in Fig. 4. This required the extension of two interfaces: Benchmark to enable dynamic task rescheduling, and DGraph to permit the dynamic modification of the graph. We also used controlled invasive accesses to inject new code for coordination that corresponds to the identification of the precise context in which exceptional situations have to be handled by means of a sequence of events, and then a farmGather composition operator (cf. 5). Fig. 7 illustrates the resulting compositional algorithm (which gives a high level view of the patterns involved to modularize the task scheduling functionality): when a benchmarking execution fails (represented by the dashed circle) the exception is matched, and an execution of a farm pattern that sends a request to all successor nodes is triggered. We then use a gather pattern to collect the availability and load average information of all successor nodes and proceed to select the best available node (bold node in the figure) and reschedule the computation.

Concretely, using our AWED implementation this is performed using the rescheduling aspect shown in Fig. 4, lines 4–11, we first identify exceptional situations as a context in which a correct initialization (state CONFIG) and the start of a concrete benchmark (START) is followed by a relevant exception (EXCEPTION) to the method Benchserver.PutArcData. We then get the list of successor nodes (state LOOKUP) that are admissible by the topology of the grid application as defined by the user through the class Nodemanager. At that point, the second advice is applied (lines 18–27) that applies the farmGather composition in order to choose the best alternative and invasively modify the graph topology by a call to the method adapter.updateGraphDefinition. Finally, we restart the benchmarking operation proper (RESTART).

The implementation of NASGrid consists of 20490 LOC. The task rescheduling concern is implemented as a whole module using the concepts of invasive operators using 391 lines of code that correspond to the TaskReschedulingAspect in AWED (97 LOC) and three auxiliary classes DFGAdapter, NodeManager and TaskUtility (294 LOC). These classes perform the dynamic graph manipulation and manage the task relocation to the new nodes. Overall we therefore achieve a concise, fully modularized and compositional implementation of the extension of NASGrid by task rescheduling, Furthermore, the composition shown in Fig. 7 provides very precise control on the contexts in which invasive modifications are performed and thus enable, in principle, to model check properties over the event sequences defining such compositions (this is however future work).

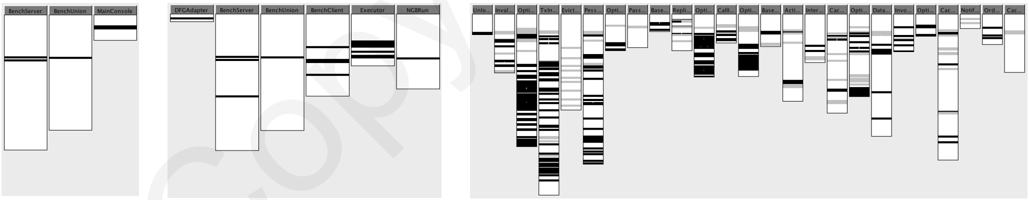
Finally, note that the overhead of task rescheduling basically consists, for each exceptional situation, in 1. A sequence of a small number of locally executed instructions up to the exceptional situation, followed by 2. A small number of parallel executions of sequences of two message exchanges for the farmGather composition and 3. a small number of local instructions to reschedule the benchmark). This overhead is clearly negligible compared to the execution of the benchmark itself in almost all use cases (*i.e.*, unless exceptional situations abound, a case that should very rarely constitute a reasonable application of NASGrid).

3.2 Degrees of Invasive Composition

In previous work we have applied invasive patterns (without support for pattern composition as introduced here) to two other evolution scenarios, an extension of NASGrid for checkpointing and an extension of the replication strategy of JBoss Cache, an infrastructure for replication under transactional control that is part of the JBoss Application Server.

Checkpointing in NASGrid. We have shown how a different reliability property of NASGrid can be improved upon via checkpointing introduction (Benavides L. et al, 2008). A checkpointing algorithm for error recovery defines a protocol to create checkpoints (snapshots of the distributed states), and guarantees the global consistency by returning to a previously-recorded state in case of failure. As we have to only add the snapshot creation and recovery actions, the degree of invasive access required is limited. It is restricted to just the context definition that triggers the snapshot creation, and also includes invasive access to the unexposed data structure in order to create its backup and play it back as part of the recovery.

Evolution of the JBoss Cache replication strategy. We have also shown how to extend the replication strategy of JBoss Cache, an infrastructure that replicates data within a cluster of distributed nodes (Benavides L. et al, 2007). The cache ensures that data replication is consistent with the transactional control over independent accesses to a distributed database. The replication and transaction functionalities are heavily crosscutting within the JBoss Cache implementation (accounting for more than 500 LOC in total scattered over a code base of around 50 KLOC). This refactoring scenario required a high level of invasive access but both concerns, replication and transactional behavior has been fully modularized using invasive patterns (once again without explicit pattern compositions).



a) NASGrid checkpointing

b) NASGrid Task rescheduling

c) JBoss Cache transactional replication

Figure 8. Crosscutting diagrams for the three evolution case studies

Fig. 8 shows the degrees of crosscutting diagrams for the three examples. Since we have been able to perform all three evolutions in a fully modular way using invasive patterns, this provides solid evidence that our approach scales from applications that are using limited invasiveness and crosscutting to applications with concerns that are highly invasive and crosscutting.

To conclude the discussion of applications of our approach, we briefly discuss if and how we can exploit the additional control we provide through the composition language introduced in this paper. Let us consider the checkpointing introduction and replication strategy extension scenarios. In the following we briefly describe these extensions and compare how the approach presented here can improve on the previous solutions. First, the pattern applications used in the NASGrid checkpointing evolution can straightforwardly be expressed using our pattern composition language and the resulting additional control would allow, *e.g.*, to reason over the correctness properties of checkpoint-based recovery (which is, admittedly rather simple in this case anyway due to the limited invasive nature). In the case of the JBoss Cache replication evolution, the

Copy Rights

additional control provided by our composition language is crucial in order to provide crucial correctness properties, such as the absence of certain race conditions during replication. This is also future work.

4. RELATED WORK

Our work is mainly related to three types of work: *(i)* other approaches to gray-box composition of software entities, *(ii)* approaches that use aspects in order to invasively manipulate software entities, mostly components, and *(iii)* work that advocates the use of patterns for distributed programming. None of these approaches, however, provides such flexible compositions of patterns for invasive composition that can be controlled precisely using a composition language. Because of space constraints we only discuss a few most relevant works. The probably best-known analysis of gray-box composition and an approach relying on code entities with holes as basic building blocks has been presented by (Aßmann U, 2003). Composition can be controlled by standard abstraction mechanisms such as component parameterization. This approach is however less structured than explicit pattern compositions and supports less precise control than ours. (Lorenz D et al, 2003) present a model for so-called aspectual collaboration in which aspects can be used to invasively modify software entities, mostly classes. Such aspect-based approaches provide no support in form of compositions of basic entities we do, and support only very coarse-grained control over invasive modifications. Patterns for distributed programming and massively programming (Cole M, 1989), (Schmidt D, 1996) have been mostly used as design patterns for non-invasive programming. Our previous work on invasive distributed patterns provide patterns as programming abstractions that can be composed manually but without support of a flexible composition language.

5. CONCLUSION AND PERSPECTIVES

In this paper e have made the case for more expressive and structured means for flexible gray-box compositions of distributed software systems. We have introduced a kernel language for structured flexible gray-box composition that enables to concisely define and precisely control pattern composition. We have sketched an aspect-based implementation of this language and applied it to a non-trivial extension of the NASGrid system for grid benchmarking. Finally, we have provided evidence that our approach scales from moderately to highly crosscutting applications. This work paves the way to the investigation of a (formally-defined) theory of invasive composition. In the long term, the quest for a set of operators that is complete with respect to a large number of evolution scenarios should be undertaken.

Copyrights

REFERENCES

Aßmann U, 2003. *Invasive Software Composition*. Springer Verlag, New York. USA

AWED website, 2010. *AWED home page* [online] available at http://awed.gforge.inria.fr (Accessed on: August 7, 2010).

Benavides L. et al, 2008. Aspect-based patterns for grid programming. *Proceedings of the 20th International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD'08)*. IEEE Press.

Benavides L. et al, 2007. Invasive patterns for distributed programs. *Proceedings of the 9th International Symposium on Distributed Objects,Middleware, and Applications (DOA'07)*. Vilamoura, Algarve, Portugal.

Benavides L. et al, 2006. Explicitly distributed AOP using AWED. *Proceedings of the 5th ACM Int. Conf. on Aspect-Oriented Software Development (AOSD'06).*

Cole M, 1989. *Algorithmic skeletons: structured management of parallel computation*. Pitman.

Frumkin R. et al, 2001. Nasgrid benchmarks: a tool for grid space exploration. *High Performance Distributed Computing*, pp. 315–322.

Kiczales G, 1996. Aspect oriented programming. *Proceedings. of the International. Workshop on Composability Issues in Object-Orientation (CIOO'96) at ECOOP*. Heidelberg, Germany.

Lorenz D et al, 2003. Aspectual collaborations: Combining modules and aspects. *The Computer Journal*, Vol. 46 No. 5. pp 542–565.

Schmidt D, 1996. OO design patterns for concurrent, parallel, and distributed systems. *Proceedings of the Second USENIX Conference on Object-Oriented Technologies and Systems (COOTS)*. Toronto, Canada

AN HYBRYD APPROACH FOR MODELS COMPARISON

Samia Benabdellah Chaouni*, Mounia Fredj* and Salma Mouline**
**ENSIAS, Mohammed V Souissi University, Rabat, Morocco*
***FSR, Faculty of Sciences, Rabat, Morocco*

ABSTRACT

Model integration problem occurs during the integration of enterprise information systems. Models comparison is an essential step of the integration, and has been discussed in several domains and various models. However, previous approaches have not correctly handled the semantic comparison. In the current paper, we develop a comparison hybrid approach which takes into account the syntactic, semantic and structural comparison aspects. We provide a rule-based system for models comparison. For this purpose, we use a domain ontology as well as other resources such as dictionaries.

KEYWORDS

UML models comparison, Ontology, rules, metamodel

1. INTRODUCTION

The information systems domain has changed dramatically in recent years under the influence of organizational evolution. This evolution can be of intern origin, generated by the restructuring of organizations, creation of new subsidiaries or new geographic or changes in business activity. Result of these factors, new information systems with their business models are created, the need to integrate existing models to make them communicate and cooperate. This evolution may also be of external origin, explained by the evolution of two organizations with the same activity domain who want merge. In this case, it must merge their information systems and more specifically their models. The goal is to integrate these models easily and efficiently.

Integration has been treated by several authors, for several models in different fields and contexts: the schemas database integration (Spaccapietra and al., 94) and (Navathe and al., 86); integration of meta-models independent models (UML, database schema, ...) (Haddar, 02) and (Pottinger and al., 03); views models integration (Anwar and al., 07) and (Rubin and al., 08); partial UML class diagrams integration (Boronat *et al.*, 06), aspect-oriented UML models integration (Ferut, 06), (Quintian, 04), (Reddy *and al.*, 06), (Lahire and al., 06), (Olivier and al., 07) and (Fleurey *and al*,. 07) ; and finally, ontology integration, which has been treated in (Falquet *et al.*, 04), (Ouagne *et al.*, 05) (Dorion *et al.*, 07) and (Bouras et al., 07). We are interested in our case in the UML models integration and more specifically the UML class diagrams (OMG UML, 09). After the analysis of models integration existing works, we found that semantic integration is a crucial problem. So far, this problem is still not properly treated. In this paper, we focus on models' comparison (the first stage of the integration process). We propose an hybrid approach which compares models syntactically, semantically and structurally. For that, we use domain ontology and other resources. The analysis of models integration existing work, we found that semantic integration is a crucial problem. So far, this problem is still not properly treated.

This article is organized as follows: Section 2 is an introduction to the general approach of integration of models. We mention in section 3 related works and their limitations. Our ontology-based proposal is developed in section 4. Some research perspectives are finally developed in the conclusion section.

Copy Rights

2. MODELS INTEGRATION

The integration is defined as the combination of components in such a way as to form a new set constituting a unit for creating synergy (Weston, 1993). Existing research (Batini and al., 86) (Pottinger and al, 03) has shown that models integration process involves two steps : 1) ***the comparison step*** is based on a set of rules called correspondence rules, also called comparison rules, mapping rules or matching rules which identify the correspondence between elements of models (correspondences created during this step are stored in a separate model called correspondence model or mapping model) ; 2) ***the integration step*** integrates models mapped in the previous step. The integration strategy relies on rules that define which and how elements will appear in the result model. These rules are (1) rules for merging the matching elements (merging rules), and (2) rules for incorporating elements that do not belong to the mapping model (rules of integration).

3. RELATED WORK

Several studies have proposed models comparison. The authors (Manning, 99), (Haddar and al., 02) and (Oliveira, 2009) provided a comparison of meta-model independent models. Databases comparison has been treated in (Madhavan and al, 01) and (Reddy and al., 06). The authors provided a comparison of UML class diagrams oriented aspects. In (Anwar and al.,07), a comparison of views models is proposed. (Uhrig et al., 2008) develop a method to compare UML class diagrams. The specification of UML 2.1 (OMG UML, 09) defines the comparison of packages.

We found different approaches of models comparison:

- Syntactic approaches: they compare the letters of strings of models elements.
- Semantic approaches: they compare the meaning associated with the compared items.
- Local structural approaches: they compare the components of the elements. For example, the comparison of local structure of two classes corresponds to the comparison of their attributes and operations.
- Global structural approaches: compare elements in relation with the elements in question. For example, the comparison of global structure of two relationships corresponds to the comparison of the two classes they connect.
- Hybrid approaches: combine two, three or four types of comparison (syntactic, semantic, global structure and local structure).

Let M1 and M2 be two models to compare. Most approaches compare **syntactically** models elements. However, they only test **identity** of elements. (Madhavan and al., 01) also detects other correspondences such as **abbreviation** (e.g. “Qty” in M1 and “Quantity” in M2) and the **acronym** (e.g. "UOM" in M1 and "UnitOfMeasure" in M2). Moreover, most approaches structurally (local and global structure) compare the models elements. Finally, all these works do not take into account the semantic aspect and are limited to detection of synonyms (e.g. "Book" in M1 and "Work" in M2) and homonyms (e.g. two classes "Family" (products) and "Family" (people)).

Our review showed on the one hand that existing works do not detect semantic mappings such as **disjunction** (e.g. two boolean attributes "Single" and "Married") and **reverse** (e.g. the relation "Buy" is the inverse of "BoughtBy” relation). Syntactic correspondences such as inclusion syntactic (e.g. “Student” and “Students”) and multilingual (e.g. “Nom” (In French) and “Name” (In English)) are not detected either. Any approach is incomplete. One may also emphasize that approaches are complementary, even though their union does not cover all types of comparison and does not detect all matches (correspondences).

On the other hand, syntactic approaches are limited because they do not detect elements that are syntactically identical but do not have the same meaning (case of homonyms) and elements which are syntactically different but which have the same meaning (case of synonyms). In addition, non-semantic approaches are limited because they do not detect elements that are syntactically different but semantically identical. Non-local structural approaches are also limited because they do not detect elements which are syntactically identical but different in local structure (e.g. two classes having the same name and no attribute in common). Finally, non-global structural approaches are limited because they cannot detect elements that are syntactically different and equivalent in global structure (e.g. two relations that are syntactically different but connect two equivalent classes).

Copy Rights

Therefore, our goal is to provide an hybrid approach incorporating syntactic, structural and semantic aspects in order to detect any mapping or correspondence.

4. PROPOSITION

Our proposal is based on ontological techniques. We therefore briefly introduce ontology concepts, before developing our approach.

4.1 Ontology

Ontologies are introduced as an"explicit specification of a conceptualization" (Gruber, 93). Domain ontologies are ontologies which are built on a particular knowledge domain. Many domain ontologies exist such as MENELAS (medical domain) (Zweigenbaum and al., 94) and TOVE (business management domain) (Gruber, 95).The domain ontology is a semantically rich model (it can express equivalence, inverse, disjunction, symmetry, transitivity, etc.), and is defined as an exhaustive list of concepts and relations between these concepts describing a particular field (Medicine, Business, Library, Restaurants, etc.).

We use an OWL ontology (Ontology Web language) because it is a W3C recommendation (Smith and al., 2004), and the meta-model OWL was defined by Ontology Definition Metamodel specification (ODM, 08) of OMG[1]. An ontology comprises the notion of "concept", also called class, corresponding to the abstractions of the relevant field. It has a name and is characterized by data properties. "Data property" allows to represent the relationship that connects the concept to a data type (integer, boolean, etc.). It is equivalent to an attribute of classe. Relationship between concepts, called "Object property", reflects the interaction between concepts, it has a name and connects a source concept called "Domain" to a target concept called "Range". "Subsumption relations", links a specific class to a more generally class.

4.2 Comparison Approach

Our goal is to provide a semantic comparison approach integrating syntactic and structural aspects as well (Figure 1). We propose a system called COM^{2}Model (Complete Comparison of Models) that takes two models as input and gives correspondence models as output. COM^{2}Model is syntactic, semantic and structural rules-based. It detects mappings between models elements. We used strategies based on semantic properties to take into account the semantic aspect. Therefore, our system refers to a domain ontology that will enable to provide semantic relevant information and decision-making during the comparison. Our system is also based on other resources to complete syntactic comparison. We use a multilingual dictionary (translation) as EuroWordNet[2], an acronym dictionary[3], an abbreviation dictionary[4], and a dictionary of synonyms as WordNet[5]. In our approach, we consider that we have at our disposal the domain ontology and the other resources. We provide a system for decision support. Our system allows the user to validate or delete mappings automatically created.

Copy Rights

[1] www.omg.org
[2] http://www.illc.uva.nl/EuroWordNet/
[3] http://acronymes.info/
[4] http://theleme.enc.sorbonne.fr/dico.php
[5] http://wordnet.princeton.edu/

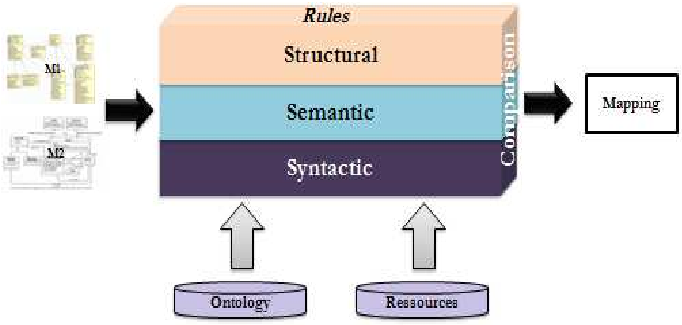


Figure 1. COM^{2}Models architecture

Our comparison process starts with the the comparison of syntactical and semantical elements (first classes, second attributes, third operations and fourth relations). It next compares elements (in the same order as just described) in global structures and in local structures.

4.3 Comparison Rules

We provided a first version of rules comparison in informal (natural) language in (Benabdellah et al., 10a) and an improved version applied to a case study in (Benabdellah et al., 10b). To specify the language for expressing these rules, we propose a meta-model.

4.3.1 MDE

Model-driven engineering1 (MDE) is a software development approach that has the potential to address the identified challenges of software engineering. It offers an environment that ensures the systematic and disciplined use of models throughout the development process of software systems. The essential idea of MDE is to shift the attention form program code to models. This way models become the primary development artifacts that are used in a formal and precise way.

The MDE approach identifies tools and materials necessary for the implementation of its paradigm. We find among others model, metamodel, language.

The most comprehensive definition of model is given by (Bézivin et al., 01): "A **model** is a simplification of a system built with an intended goal in mind. The model should be able to answer questions in place of the actual system." According to (MOF, 02), "A **metamodel** is a model that defines the **language** for expressing a **model**".

In our case, the model is the comparison rules. We define a metamodel that defines the language for expressing these rules.

Copyrights

4.3.2 Rules Metamodel

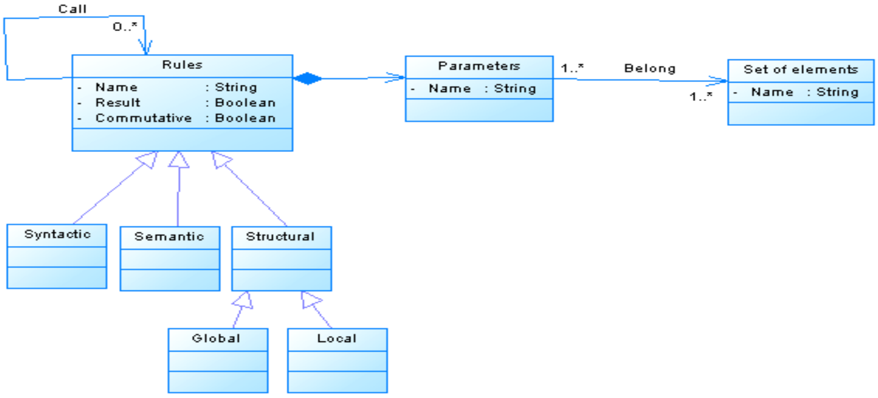


Figure 2. Comparison rules metamodel

We modeled our metamodel in UML language. The rule is characterized by a name, a boolean result (i.e. true or false) and the type (commutative or not). The rule can be syntactic, semantic, global structure or local structure. It is composed of parameters that have a name. These parameters belong to a set of elements. A rule can call one or more other rules.

4.3.3 Comparison Rules Examples

We first established the syntactic comparison rules: rule of identity, rule of inclusion, rule of equivalence multilingual, rule of acronym, rule of abbreviation and rule of syntactic equivalence. Then the comparison semantic rules : rule of synonymy of classes, rule of equivalence of classes (as an ontology), rule of semantic equivalence of classes, rule of hyponymy of classes, rule of synonymy of attributes, rule of disjunction of attributes, rule of semantic equivalence of attributes, rule of operations synonymy, rule of semantic equivalence of operations, rule of synonymy of relations, rule of inverse relation , rule of equivalence of relations (as an ontology), and rule of semantic equivalence of relations. Then the rules for comparing global structure elements (classes, attributes, operations, relations and generalization relation). And finally, rules for comparing local structure elements (classes, attributes, operations and relations).

Some representative rules in accordance to the comparison rules metamodel are described below.

- **Rule of syntactic inclusion of two elements elt$_i$ and elt$_j$**

This is a syntactic rule, called "Syntactic_inclusion", compares two elements (parameters) called D_1elt_i and D_2elt_j. The first element belongs to the set of elements of the first diagram called D_1E and the second element belongs to the set of elements of the second diagram called D_2E. This commutative rule returns 1 (true) if the first elements are included syntactically in the second, and else returns 0 (false).

$$\text{Syntactic\_inclusion} : D_1E \text{ x } D_2E \rightarrow \{0,1\}$$
$$\text{Syntactic\_inclusion}(D_1elt_i, D_2elt_j) = \begin{cases} 1, & \text{if } \exists\, p, s \in \S \mid D_1elt_i.name = p + D_2elt_j.name + s \text{ ou } D_2elt_j.name = p + D_1elt_i.name + s \\ 0 & else \end{cases}$$

Rule explanation: A first element is syntactically included in a second element if the name of the first element appended to a prefix and (or) a suffix gives the name of the second element.

- **Rule of semantic equivalence of two relations R$_i$ and R$_j$**

This is a semantic rule, called "Equivalence_semantic_relations", compares two elements (parameters) called D_1R_i and D_2R_j. The first element belongs to the set of relations of the first diagram called D_1R and the second element belongs to the set of relations of the second diagram called D_2R. This rule called other rules called "Synonymy_elements", "Inverse_relations" and Equivalence_Ontologie_relations". This commutative rule returns 1 (true) if the two elements are semantically equivalent, and else returns 0 (false).

$$\text{Equivalence\_semantic\_relations}: D_1R \text{ x } D_2R \rightarrow \{0,1\}$$
$$\text{Equivalence\_semantic\_relations}(D_1R_i, D_2R_j) = \begin{cases} 1, si & \text{Synonymy\_elements}(D_1R_i, D_2R_j) = 1 \text{ or Inverse\_relations}(D_1R_i, D_2R_j) = 1 \\ & \text{or Equivalence\_Ontologie\_relations}(D_1R_i, D_2R_j) = 1 \\ 0, & else \end{cases}$$

Rule explanation: Two relations are semantically equivalent if they are equivalent (in reference to ontology) or reverse.

- **Rule for comparing global structure of two relations R$_i$ and R$_j$**

This is a global structural rule, called "Equivalence_structure_global_relations", compares two elements (parameters) called D_1R_i and D_2R_j. The first element belongs to the set of relations of the first diagram called D_1R and the second element belongs to the set of relations of the second diagram called D_2R. This rule calls other rules. This commutative rule returns 1 (true) if the two elements are equivalent in global structure, and else returns 0 (false).

Copy Rights

$$\text{Equivalence\_structure\_global\_relations } D1R \text{ x } D2R \rightarrow \{0,1\}$$

$$\text{Equivalence\_structure\_global\_relations}(D_1R_i, D_2R_j) = \begin{cases} 1, \mathit{if}\ [(\exists D_1C_k, D_1C_m \in D_1C, \exists D_2C_l, D_2C_n \in D_2C \mid D_1R_i(D_1C_k, D_1C_m) \text{et } D_2R_j(D_2C_l, D_2C_n)) \\ \mathit{and}\ (\text{Equivalence\_semantic\_classes}(D_1C_k, D_2\ C_l) = 1\ \mathit{or}\ \text{Equivalence\_syntactic\_elements}(D_1C_k, D_2\ C_l))] \\ \mathit{and}\ (\text{Equivalence\_semantic\_classes}(D_1C_m, D_2\ C_n) = 1\ \mathit{or}\ \text{Equivalence\_syntactic\_elements}(D_1C_m, D_2\ C_n))] \\ \text{Or } [\exists D_1C_k, D_1C_m, D_1C_o \in D_1C, \exists D_2C_l, D_2C_n \in D_2C, \exists D_1G_p \in D_1G \mid D_1R_i(D_1C_o, D_1C_m), D_2R_j(D_2C_l, D_2C_n) \text{ and} \\ D_1G_p.\text{super\_class} = D_1C_o \text{ et } D_1G_p.\text{sub\_class} = D_1C_k\ \mathit{and}\ (\text{Equivalence\_semantic\_classes}(D_1C_k, D_2\ C_l) = 1 \\ \mathit{ou}\ \text{Equivalence\_syntactic\_elements}(D_1C_k, D_2\ C_l))]\ \mathit{and}\ (\text{Equivalence\_semantic\_classes}(D_1C_m, D_2\ C_n) = 1 \\ \mathit{or}\ \text{Equivalence\_syntactic\_elements}(D_1C_m, D_2\ C_n))] \\ 0, & \mathit{else} \end{cases}$$

Rule explanation: Two relations D_1R_i and D_2R_j are equivalent in global structure if: [There is two classes D_1C_k and D_1C_m such as D_1R_j links it and there is two classes D_2C_l, D_2C_n such as D_2R_j links them and D_1C_k and D_2C_l are syntactically or semantically equivalent] Or [There is two classes D_1C_k and D_1C_m and there is D_1C_o class such as D_1C_o is the super class of D_1C_k and D_1R_i links D_1C_o and D_1C_m and there is two classes D_2C_l, D_2C_n such as D_2R_j links them and D_1C_k and D_2C_l are syntactically or semantically equivalent and D_1C_m and D_2C_n are syntactically or semantically equivalent]

- **Rule for comparing local structure of two classes C_i and C_j**

This is a local structural rule, called “Equivalence_structure_local_classes”, compares two elements (parameters) called D_1C_i and D_2C_j. The first element belongs to the set of classes of the first diagram called D_1T and the second element belongs to the set of classes of the second diagram called D_2T. This rule calls other rules. This commutative rule returns 1 (true) if the two elements are equivalent in local structure, and else returns 0 (false).

$$\text{Equivalence\_structure\_local\_classes: } D_1C \text{ x } D_2C \rightarrow \{0,1\}$$

$$\text{Equivalence\_structure\_local\_classes}(D_1C_i, D_2C_j) \\ = \begin{cases} 1, \mathit{if}\ (\forall D_1C_iT_k \in D_1C_iT, \exists D_2C_jT_l \in D_2C_jT \mid \text{Equivalence\_semantic\_attributes}(D_1C_iT_k, D_2C_jT_l) = 1 \\ \mathit{or}\ \text{Equivalence\_syntactic\_elements}(D_1C_iT_k, D_2C_jT_l))\ \mathit{and}\ (\forall D_2C_jT_l \in D_2C_jT, \exists D_1C_iT_k \in D_1C_iT \mid \\ \text{Equivalence\_semantic\_attributes}(D_1C_iT_k, D_2C_jT_l) = 1\ \mathit{or}\ \text{Equivalence\_syntactic\_elements}(D_1C_iT_k, D_2C_jT_l)) \\ \mathit{and}\ (\forall D_1C_iOP_k \in D_1C_iOP, \exists D_2C_jOP_l \in D_2C_jOP \mid \text{Equivalence\_semantic\_attributes}(D_1C_iOP_k, D_2C_jOP_l) = 1 \\ \mathit{or}\ \text{Equivalence\_syntactic\_elements}(D_1C_iOP_k, D_2C_jOP_l))\ \mathit{and}\ (\forall D_2C_jOP_l \in D_2C_jOP, \exists D_1C_iOP_k \in D_1C_iOP \mid \\ \text{Equivalence\_semantic\_attributes}(D_1C_iOP_k, D_2C_jOP_l) = 1 \mathit{or}\ \text{Equivalence\_syntactic\_elements}(D_1C_iOP_k, D_2C_jOP_l)) \\ 0, & \mathit{else} \end{cases}$$

Rule explanation: Two classes are equivalent in local structure if their attributes and operations are syntactically or semantically equivalent.

5. CONCLUSION

Any approach to model comparison must take into account syntactic, semantic and structural aspects. The semantic integration of models is a complex task because it requires understanding the semantics of linking concepts. The main contribution of this paper is to compare syntactic, semantic and structural aspects of two models. The development of our application is done in Java because this language allows the use of several APIs for manipulating OWL ontologies as Jena (http://jena.sourceforge.net/) and Sesame (http://jena.sourceforge.net/). Other resources (dictionaries) are managed in tables. We are currently achieving an interactive user interface of our system. In fact, the user validates or delete mappings created. Validated correspondences will be stored in a MySQL database. The integration can be applied to "n" models $M_i=\{Mi|\ i=1..n\}$. In this case, we can integrate M_1 and M_2, then integrate their result model $MR_{1.2}$ and M_3, etc.., until the M_n model. Our research will be a further study on the definition of rules of integration and merger, which will thus enable to realize the whole process of model integration.

Copyright

REFERENCES

Anwar, A., Ebersold, S., Coulette, B., Nassar, M. and Kriouile, A., dec 2007. Vers une approche à base de règles pour la composition de modèles. Application au profil VUML," L'Objet, Hermès Science Publications, Numéro spécial Ingénierie Dirigée par les Modèles, Vol. 13, N. 4/2007, p. 73-103

Batini, C. Lenzerini, M. Navathe, S.B, dec 1986. A Comparative Analysis of Methodologies for Database Schema Integration. ACM Computing Surveys, 18(4):323–364.

Benabdellah, C. S. and Fredj, M. 2010.Vers une contribution à l'intégration sémantique des modèles UML. ERATSI, INFORSID, France (Benabdellah and al., 2010b)

Benabdellah, C. S., Fredj, M., Mouline, S., 2010. Un système à base de règles d'aide à la décision pour la comparaison sémantique des modèles", SITA'2010, Rabat (Benabdellah and al., 2010a)

Bézivin, J., Gerbé, O. Nov 2001. Towards a Precise Definition of the OMG/MDA Framework. ASE'01

Boronat, A., Carsí, J.A., Ramos, I. and Letelier, P., 2007. Formal Model Merging Applied to Class Diagram Integration. Electronic Notes in Theoretical Computer Science (ENTCS).

Bouras, A., Gouvas, P. and Mentzas, G. 2007. ENIO: An Enterprise Application Integration Ontology. 18th International Workshop on Database and Expert Systems Applications.

Dorion, E. and Fortin, S. 2007. Semantic Interoperability: Revisiting the Theory of Signs and Ontology Alignment Principles. 12th International Command and Control Research and Technology Symposium "Adapting C2 to the 21st Century".

Falquet, G., Jiang, C. M. and Ziswiler, J. 2004. Intégration d'ontologies pour l'accès à une bibliothèque d'hyperlivres virtuels. Actes du 14ème Congrès Francophone de Reconnaissance des Formes et Intelligence Artificielle (RFIA2004), Toulouse, France.

Ferut, T., June 2006. Définition d'un mécanisme de composition appliqué aux modèles métiers, Stage du Master PLMT, Laboratoire I3S, équipe OCL.

Fleurey, F., Baudry, B., France R. and Ghosh, S. 2007. A Generic Approach For Model Composition" Proceedings of the Aspect Oriented Modeling. Workshop at Models 2007, Nashville USA.

Gruber, R.T. 1995. Towards Principles for the Design of Ontologies Used for Knowledge Sharing, International Journal of Human Computer Studies, Vol. 43, (No 5/6), pp. 907-928.

Gruber, T., 1993. A Translation Approach to Portable Ontology Specifications, Knwledge Acquisition, Vol. 5 (No. 2) pp. 199-220.

Haddar, N., Gargouri, F. and Ben Hamadou, A., 2002. Une approche formelle pour l'intégration des aspects structuraux et comportementaux de représentations conceptuelles ISDM Journal, N°19.

Lahire, P. and Quintian, L., 2006. New perspective to improve reusability in object oriented languages. Journal Of Object Technology (JOT), 5(1) pp. 117_138.

M. K. Smith, C. Welty and D. L. McGuinness, OWL Web Ontology Language – Guide, http://www.w3.org/TR/2004/REC-owl-guide-20040210/

Manning, C. and Schutze, H., 1999. Foundations of Statistical Natural Language Processing, ISBN 978-0262133609, MIT Press.

MOF OMG, April 2002. Meta Object Facility (MOF) Specification. Version 1.4.

Navathe, S. B., Elmasri, R. and Larson, J., 1986. Integrating user views in database design, IEEE Computer 19,l (Jan.), pp. 50-62.

Oliveira, K. et Breitman, K., 2009. A Flexible Strategy-Based Model Comparison Approach: Bridging the Syntactic and Semantic Gap. Journal of Universal Computer Science, vol. 15, no. 1.

Olivier, B., Philippe, L., Alexis, M., Noël, P., and Gilles, V.. 2007. Evaluation de l'apport des aspects, des sujets et des vues pour la composition et la réutilisation des modèles", Revue RSTI-L'Objet, 13 (2-3), pp. 177-212.

OMG Unified Modeling Language (OMG UML), 2007. Infrastructure, Version 2.2, OMG Document Number: formal/2009-02-04 Standard document.

Ontology Definition Metamodel, September 2008, OMG Adopted Specification, OMG Document Number: ptc/2008-09-07, http://www.omg.org/docs/ptc/08-09-07.pdf

Ouagne, D., Le Bozec, C., Zapletal, E., Thieu, M. and Jaulent, M. 2005. Intégration de multiples ontologies en pathologie mammaire. Journées Francophones d'Informatique Médicale, Lille.

Quintian, L. , 2004. JAdaptor : Un modèle pour améliorer la réutilisation des préoccupations dans le paradigme objet, Thèse de doctorat, Université de Nice-Sophia Antipolis, France.

Rachel, A. Pottinger and Philip, A. Bernstein. 2003. Merging Models Based on Given Correspondences. Technical Report UW-CSE-03-02-03, University of Washington.

Copy Rights

Reddy, Y.R., Ghosh, S., France, R.B., Straw, G., Bieman, J. M., McEachen, N., Song, E. and Georg, E. 2006. Directives for Composing Aspect-Oriented Design Class Models, T. Aspect-Oriented Software Development, pp.75-105.

Rubin, J., Chechik, M. and Easterbrook, S. M. 2008. Declarative approach for model composition. Proceedings of the 2008 international workshop on Models in software engineering, May 10-11, Leipzig, Allemagne.

Spaccapietra, S., Parent, C. and Dupont, Y., April 1994. View Integration : a step forward in solving structural conflicts. IEEE Transactions on Data and Knowledge Engineering, vol. 6, no.2.

Uhrig, S., 2008. Matching Class Diagrams: With Estimated Costs Towards The Exact Solution?, Workshop on Comparison and Versioning of software Models, ACM.

Weston, R. H., 1993. "Steps towards enterprise wide integration: a definition of needs and first generation open solutions". International Journal of Production Research, 31(9), 2235-2254.

Zweigenbaum, P. 1994. MENELAS : An Access System for Medical recods Using Natural Lnguage, Computer Methods and Programs in Biomedecine, Vol. 45, pp. 117-120.

Copy Rights

ADAPTIVE PRE-PROCESSING OF LARGE POINT CLOUDS FROM OPTICAL 3D SCANNERS

Erik Trostmann, Christian Teutsch and Dirk Berndt
FraunhoferInstitute for Factory Operation and Automation (IFF)
Sandtorstrasse 22, D-39106 Magdeburg, Germany

ABSTRACT

Optical 3d scanners have a wide distribution in industrial applications, mainly to digitize object surfaces in order to measure and compare geometric properties. But the generated 3d data is far from being optimal. They contain noise, outliers and artifacts which requires a pre-processing step. We present a set of fast and adaptive methods for 3d point cloud optimization which are suitable for industrial measuring tasks. We discuss appropriate data structures and efficient algorithms for nearest neighbor queries in large point clouds as well as curvature based simplification and smoothing approaches.

KEYWORDS

3d-scanner, point cloud optimization, simplification, smoothing

1. INTRODUCTION

In contrast to digital geometric modeling, the data obtained from optical 3D scanners usually contain several errors caused by system and measuring principle specific characteristics. Additionally, the data is affected by environmental influences such as unfavorable lighting conditions, dust or vibrations. Measurements on the generated data may lead to incorrect results. Instead of applying mesh optimizations we prefer to pre-process the scan data before a mesh is constructed. Therefore, procedures for the point cloud optimization, evaluation and inspection are needed that are robust against external influences. While a smooth and aesthetic visualization is desired for most computer graphic applications, our application field is industrial measurements, which primary require fast and automated procedures with a high reliability.

When analyzing 3D scan data usually very large data sets with millions of points have to be processed. Additionally, due to overlaying scans, redundant information with different quality is generated (see Figure 1). Many applications do not require such a high point density, and additionally, the computation time for the data analyses is often limited in practice (cycle times). Therefore, it is necessary to optimize the data and minimize the number of points while minimizing the loss of information at the same time. The bases for an efficient data processing are data structures which support the efficient retrieval of neighborhood information from the point set. Thus, local neighborhood information must be reconstructed from the basic 3D point coordinates.

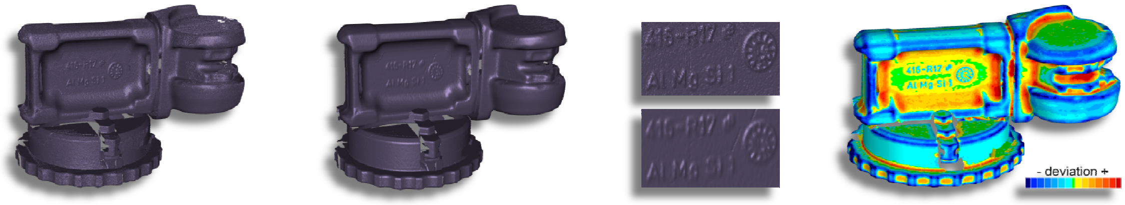


Figure 1. Meshed point cloud of an iron casting before and after having performed the pre-processing steps. The colors show the deviations to the original data set. They also indicate the positions were the proposed methods are particularly suitable for, that is, edges and small details.

Copy Rights

We discuss approaches to derive local point information from different scans and scanners. This includes tree-based data structures as well as point cloud simplification, smoothing and nearest neighbor problems. The following criteria are fulfilled by our methods. They are fast to utilize them for production. They are stable, which means reliable and very insensitive against noise, outliers, uneven sampling and artifacts. In particular, they are as adaptive as possible to reduce the numbers of thresholds, because thresholds tend to perform well only on the test samples from which they were derived. We mainly derive our scan data sets from light-section devices (e.g. fringe/laser line projection) but also from laser scanners as used for digitizing buildings.

A polygonal mesh representation mostly conceals holes and imperfect point data because it aims to produce an esthetic nice looking model (Gois et al. (2007)). But we are particularly interested in those uncertainties to evaluate the acquisition quality of the scanner and to locate deviations for measuring tasks. Furthermore, direct point cloud processing avoids the time-consuming meshing operations for large data sets, assumed that a mesh is not needed at the end, which is usually the case for measurements. But nevertheless, in this work we also employ meshes afterwards to illustrate the effect of our methods.

We also discuss approaches how existing algorithms benefit from system internals which are provided by optical 3D scanners and the underlying measuring principles, respectively.

2. POINT CLOUD ANALYSIS

Having large sets of 3D data one usually needs to know more about its spatial structure in order to perform specific operations. The most important information is about the local neighborhood which must be reconstructed from the given data.

First of all, the set of 3D coordinates must be organized and structured in order to analyze local neighborhoods between multiple scanlines and scans from different sensors, etc. Therefore, a data structure is needed, which allows to efficiently search for points within the entire point cloud. There are different algorithms available for the structuring of 3D points, based on tree and graph representations or on the clustering of data like in Park et al. (2006). One of the most important applications is the reconstruction of local neighborhoods. For example, finding the nearest or the m-nearest points to a given point is a typical task. So we first take a look on existing methods and then choose the most appropriate.

2.1 3D Data Organization and Representation

An intuitive method for structuring 3D data is partitioning, where points are assigned to a unique cluster. A popular data structure is provided by octrees, which hierarchical stores nearby points in identical cells (e.g. cubes). Mainly computer graphics applications utilize this data structure, which supports the implementation of efficient divide-and-conquer strategies like Chen (2006). For spatial nearest neighbor queries of large point clouds a more suitable structure is found by kd-trees. They are a special case of binary space partitioning (BSP) trees, whereas a point set is subdivided into axis-aligned non-intersecting cuboidal regions. The splitting plane in each recurrence is defined by the median value of the current ordinates.

To find the median of a set of values, one may proceed by sorting and selecting the central value in the array. Since sorting yields much more information than just the median, this procedure is wasteful. The fastest method for finding the median is partitioning, exactly as it was done in the quicksort algorithm. Therefore we use an algorithm for finding the m-largest element (m is the medial position) as presented by Press et al (2007). It is an in place method, that avoids data copying and thus saves time and memory. The operation count scales as $\mathrm{O}(n)$ rather than $\mathrm{O}(n \log n)$ for a complete sorting.

Because the three-dimensional kd-tree for a set of n points is a binary tree with n leafs, it only uses $\mathrm{O}(n)$ storage and the construction time is $\mathrm{O}(n \log n)$. The query algorithm only visits those nodes whose regions are properly intersected by the query range. The query time is bounded by $\mathrm{O}(n^{(1-1/k)} + p_r)$, where p_r is the number of points reported.

In practice there are two degenerate cases we must take care of. The first is the presence of identical ordinates for which no unique median position exists. A solution is given by lexicographic ordering using the values of the other ordinates. The second case is the presence of (at least two) identical points which cannot be solved even by lexicographic ordering. One of two equal points will be falsely inserted at a tree position where it cannot be found. But fortunately this has no serious impact on the nearest neighbor searches since the other point, which is the same, will be found anyway.

2.2 Nearest Neighbor Queries and the Optimal Neighborhood

To analyze the local neighborhood of a given point, nearby points in a spherical neighborhood are searched. The density ρ is derived from the radius r of the neighborhood, and the number of points inside N_p. Assuming a locally planar surface the density is computed by $\rho = N_p/(\pi r^2)$. For the examined objects, the local point density does not vary significantly. Thus, the computation performance can be increased by estimating the globally best radius as the median radius derived from only 1% of all points, provided that at least 20 points are covered.

There are two factors that mainly describe a neighborhood for further algorithms: the number of points and the quality of the local representation. Local analyses typically require a certain number of points k to estimate local shape functions (e.g. fitting geometries) depending on the number of unknown parameters. But the number of neighboring points is not necessarily a sufficient criterion, if the considered region is too small or too large and does not represent the surface part. Especially in scanned point clouds with noise and uneven sampling densities, the point number fluctuates locally. Therefore, the point number needs to be coupled to a neighborhood radius, which should be chosen adaptively. The radius range should be defined by the data density and expected sizes of the minimal and maximal features that should be analyzed.

To estimate surface normals, a plane is usually fitted to neighboring points. Therefore, Mitra et al. (2004) proposed an adaptive method to compute the optimal radius with respect to the amount of points and their distribution. Assuming a Gaussian distribution where noise has zero mean and standard deviation σ_n, they minimize a bound for the estimated angle between the normal vectors of the fitted plane and the true surface with a probability of $(1-\varepsilon)$. The optimal radius r is obtained by:

$$r = \left(\frac{1}{\kappa}(c_1 \frac{\sigma_n}{\sqrt{\varepsilon\rho}} + c_2\sigma_n^2\right)^{\frac{1}{3}} \tag{1}$$

where ρ is the local sampling density, κ is the local curvature, and c_1 and c_2 are constants. The algorithm takes σ_n as input and iteratively evaluates r. In the first iteration ρ and κ are evaluated based on empirically chosen $k(=15)$ nearest neighbors and then the radius r is obtained from Eq. (1).

Another approach is discussed by Ohtake (2003) and Nagai (2009). They locally fit 3D quadrics and bivariate quadratic functions, which exhibit more degrees of freedom and are more suitable to represent curved surface parts. Their optimal radius r is adaptively increased until sufficient k neighbors are found to solve the surface equations.

For the used 3D scanners, there is some a priori information that can be used to adaptively determine a minimum radius. The movement and displacement between two successive scanlines are known system parameters, which typically range between 0.1 and $0.5\ mm$ for the examined models. Thus, in any case, the minimum radius r_{min} must at least be larger than the distance between two scanlines Δs. Starting from a given point on a scanline, a cubic region can be defined, which touches the neighboring scanlines. Then, the circumsphere of this region contains a sufficient number of points from the neighboring scanlines. Thus, the minimum radius r_{min} of this circumsphere is defined by:

$$r_{min} = \sqrt{2}\varDelta s + c \tag{2}$$

with the constant c to compensate uncertainties (usually $c = 0.1\ mm$). For the flexible laser scanner, the point density is usually higher and the scanlines are oriented in an arbitrary manner. In this case, r_{min} is adaptively increased until at least 20 points are covered. The starting value must be defined by a value larger than the uncertainty of the scanner, which is also $c = 0.1\ mm$.

2.3 Outlier Detection

Outliers are found by coordinates in a low density environment where only a few neighbors are present in a certain distance. It mostly depends on the application which density signals outlier. We propose an adaptive

Copy Rights

procedure that computes for each point the distance to the nearest neighbor and keeps all points with distances smaller than two times the standard deviation, which is (statistically) about 95%. This procedure works well in practice but an optimization is still possible. In addition to the global one Kriegel et.al (2009) obtain an local outlier score regarded to the local densities only that enables a more subtle decision if a point distance signals an outlier or not.

2.4 Surface Normal Estimation

The local orientation of the surface is described by its normal vectors. Since the surface function is unknown in most cases, it must be approximated. Besides polygonal meshing, from which this information can directly be derived, geometry approximation techniques are usually employed. For example Ohtake (2003) locally fits 3D quadrics and bivariate quadratic functions to determine the surface normal vector.

Dey (2005) compared the different methods and found out that Delaunay triangulation is the most robust but also the significant slowest.

For the scan data in this work a plane fitting is performed, since this method is fast and robust and the point sets are dense. The plane fitting procedure is then based on a least-squares orthogonal distance fitting (ODF) as proposed by Ahn (2004). Therefore we compute the central moments tensor of the point set as the symmetric square matrix $\boldsymbol{M}$ with the mean values (X_0, Y_0, Z_0) by:

$$\boldsymbol{M} \equiv \begin{pmatrix} M_{xx} & M_{xy} & M_{zx} \\ M_{xy} & M_{yy} & M_{yz} \\ M_{zx} & M_{yz} & M_{zz} \end{pmatrix}, \quad \begin{array}{lll} x_i = X_i - X_0 & y_i = Y_i - Y_0 & z_i = Z_i - Z_0 \\ M_{xx} = \sum_{i=1}^{m} x_i^2 & M_{yy} = \sum_{i=1}^{m} y_i^2 & M_{zz} = \sum_{i=1}^{m} z_i^2 \\ M_{xy} = \sum_{i=1}^{m} x_i y_i & M_{yz} = \sum_{i=1}^{m} y_i z_i & M_{zx} = \sum_{i=1}^{m} z_i x_i \end{array} \qquad (3)$$

The matrix $\boldsymbol{M}$ is then decomposed by using the singular value decomposition (SVD) of matrices.

$$\boldsymbol{M} = \boldsymbol{V}_M \boldsymbol{W}_M \boldsymbol{V}_M^T \qquad (4)$$

As a result, the diagonal matrix $\boldsymbol{W}_M$ contains the principle central moments, and the orthogonal matrix $\boldsymbol{V}_M$ contains the principle axes of central moments. The fitting plane is finally defined by the mass center X_c and the principle axis $\boldsymbol{v}_{Mj}$ with the corresponding smallest moment $\boldsymbol{w}_M$.

Pauly (2003) observed that the geometry fitting should respect the nearby points more than the distant points. Hence, we weight neighboring points based on their distances to p by using a Gaussian function.

The orientation of the estimated normals from the plane fitting depends on the position of the plane within the coordinate system. For the analysis and for an esthetic visualization a consistent orientation is necessary. By exploiting given scanline information of the camera or the laser position, the normal vector orientation problem becomes trivial. An approximated normal n for a point p only needs to be flipped, if it points contrary to the camera or laser position (p_c/p_l), specifically:

$$n = -n \ \ if \ \ n \cdot \frac{p_c - p}{\parallel p_c - p \parallel} < 0. \qquad (5)$$

2.5 Graph Representations for Unstructured Point Sets

Without additional information a data structure is necessary that encodes the neighborhood of the points and that ensures a consistent orientation of their normal vectors. Therefore, we construct an Euclidean Minimum Spanning Tree (EMST) for the entire point set P. An edge $E(i,j)$ is added to the tree if either p_i is in the neighborhood of some p_j or vice versa. Specifically, the Euclidean minimum spanning tree of the point set P is the maximal tree $\mathrm{EMST}(P) = (P, E)$ such that $E \subseteq P \times P$ and the sum of all edge lengths $\sum l(e_k)$ is minimum, where $l(e_k) = |p_i - p_j|$. Finally, each point has a path to its nearest neighbor and the graph encodes the geometric proximity in the Euclidean norm in $\mathbb{R}^3$ (also called a Riemannian Graph). For a more detailed discussion on nearest neighborhood graphs the reader is referred to Attene et al. (2000).

Copy Rights

3. POINT CLOUD OPTIMIZATION

Repeated scans in the same area of an object's surface and the choice of the sensor alignment can cause overlaying point clouds. The introduced redundancy can be helpful to locally optimize the point cloud on the one hand. But on the other hand, a large amount of points significantly reduces the processing speed of further algorithms. This section discusses suitable methods to process the redundant information in order to optimize and simplify a point cloud with respect to quality and curvature information as derived in the chapters before.

3.1 Adaptive Smoothing

As discussed in the data acquisition section, rough and specular surfaces can cause high-frequent noise for optical sensors. Furthermore, the local point distribution variates depending on the shape and the distance to the sensors. By applying smoothing algorithms, this variance can be reduced. Overlaying scans can also produce errors which origin from an imprecise calibration of the sensors to the axes movements. For an aesthetic visualization and for more robust post-processing algorithms, a smoothing procedure must be employed. When smoothing noisy data, edges should be preserved and the quality of a point, regarding to its viewing conditions, should have a notable influence on the resulting point. The smoothing procedure locally operates in a neighborhood defined by the k-nearest neighbors (typically $k = 20$) and the minimal radius r_{min}, depending on the scanline distance Δs as proposed in the section before. As long as the number of points in the neighborhood is smaller than k, the search radius r is increased, starting from r_{min}. A smoothed point is derived from all of its neighbors by applying a weighting function that depends on the distance of the neighbor to the considered point d_i and a weight ω_i, regarding the scanline curvature κ_i and quality of the viewing conditions q_i (see Figure 2 (a)). Since a low quality can cause noise and thus a higher curvature, the weight ω_i is defined as the normalized sum of these measures by:

$$\omega_i = (1-\alpha)\bar{\kappa}_i + \alpha(1-\bar{q}_i) \qquad with \quad 0 \leq \alpha \leq 1, \tag{6}$$

where α allows to manipulate the ratio between the influence of the viewing quality (viewing and projection angles) and the edge values (typically $\alpha = 0.5$). $\bar{\kappa}_i$ and $\bar{q}_i$ are the normalized values of κ_i and q_i with:

$$\bar{q}_i = \frac{q_i}{\pi}, \qquad \bar{k}_i = \begin{cases} \frac{\kappa_i}{\tau} & \text{if } \kappa_i < \tau, \\ 1 & \text{otherwise.} \end{cases} \tag{7}$$

This normalization is based on the following observation: Since the viewing quality q_i is defined by the viewing angle, its range is limited between 0 and π. The scale τ defines the curvature value, that indicates significant edges. It is an empirical measure, and for the scanline curvature based on 4th order NURBS curves it was found out that $\tau = 0.2$ gives optimal results (Teutsch et al. (2007)).

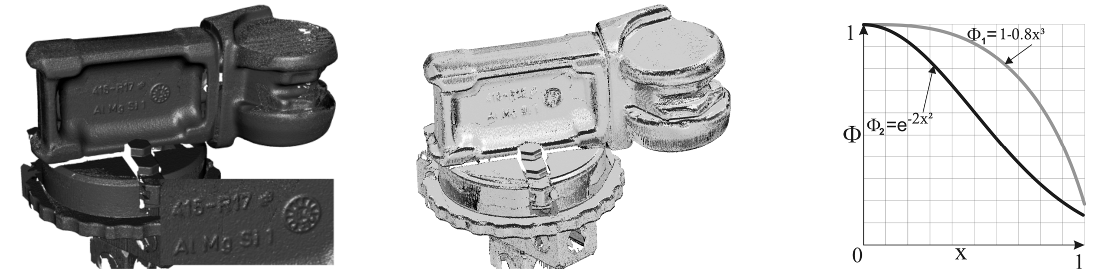
After having defined a weight for each neighbor, an influence function Φ is added. Since the number of points within the bounding sphere of the neighborhood nonlinearly increases with the radius r, the influence function should penalize points near the edge of neighborhood more than nearby points. Based on the function Φ, its neighbors p_i and their weight ω_i, the resulting smoothed point p_s is computed by:

$$p_s = \frac{1}{\omega_{sum}} \sum_{i=1}^{n} p_i \omega_i \Phi(\bar{d}_i), \quad \text{with} \quad \omega_{sum} = \sum_{i=1}^{n} \omega_i \Phi(\bar{d}_i), \tag{8}$$

where the measure $\bar{d}_i$ is the adaptively normalized distance d_i to the neighbor p_i. Since the maximum for d_i is limited by the radius r of the neighborhood, $\bar{d}_i$ is defined as:

$$\bar{d}_i = \frac{d_i}{r} \quad \text{with} \quad d_i = |p_i - p| \tag{9}$$

Copy Rights



(a) (b) (c)

Figure 2. Original point cloud (a), Illustration of the weights for single points based on their scanline curvature κ_i and the value q_i for the viewing angle (b). The smoothing effect at the edges (dark) is lower than in planar areas. The paths of two influence functions Φ are shown in (c). The empirically chosen parameters guarantee an influence between 10-20% at the edge of the neighborhood (x=1).

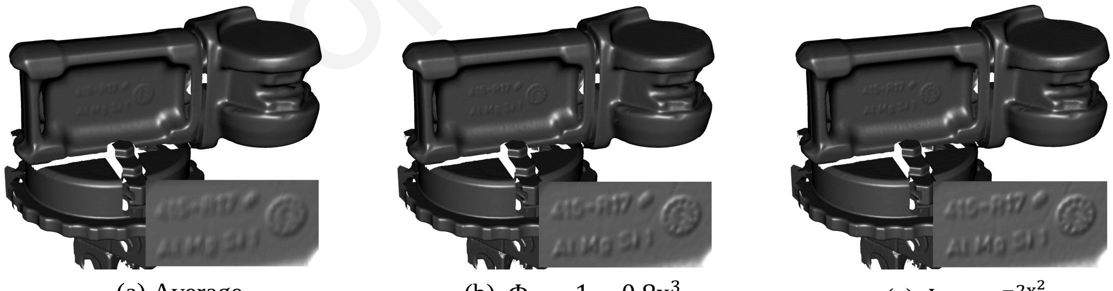
Together with the average function, two different nonlinear influence functions Φ_1 and Φ_2 were applied in order to attain the desired result. The functions and their paths are illustrated in Figure 2(c). A radius of r=1mm was chosen.

In summary the curvature and quality-based weighting with an influence function performed well and exhibited a significant improvement compared to the simple average, since edges are retained while more planar regions are smoothed. Due to its strong slope, the exponential function(Φ_2) operates more locally, and thus gives more influence to nearby points than the cubic function (Φ_1).

In order to control the smoothing, a test function is additionally applied, which checks if the distance of a smoothed point p_s to its original p exceeds a tolerance t. The value of t depends on the measuring uncertainty or the accepted inaccuracy (e.g.: $t = 0.1$):

$$p_s = p + t\frac{p_s^* - p}{\|p_s^* - p\|}. \qquad (10)$$

Without having scanline curvature, Lange et al. (2005) propose a method for point cloud fairing using an anisotropic geometric mean curvature flow. Their method solves a parabolic PDE with boundary constraints to obtain an anisotropic Laplacian operator. Unfortunately, this approach requires many iterations and a user-defined parameter called edge quotient that enables to emphasize corners. Furthermore, based on the given normal vectors of a point cloud, Nielson (2004) achieves a smoothing by generating an implicit volume model whose zero level isosurface interpolates the given points and associated normal vectors.



(a) Average (b) $\Phi_1 = 1 - 0.8\mathrm{x}^3$ (c) $\Phi_2 = \mathrm{e}^{-2\mathrm{x}^2}$

Figure 3. Triangulated models to visualize the effect of smoothing a point cloud in different stages. The smoothed representation of the original noise point cloud based on an average filtering (a). The better results by applying the influence functions Φ_i from Figure 2(b) are shown in (b) and (c).

In addition to point cloud smoothing, there are also many smoothing algorithms for polygonal meshes. These methods benefit from the known local edge connections, e.g. to relax the polygons as discussed by Bade et al. (2007). Furthermore, Nealen et al. (2006) introduce a framework for triangle shape optimization and feature preserving smoothing of triangular meshes that is guided by the uniformly weighted Laplacian and the discrete mean curvature normal. A comparative overview on polygonal mesh smoothing is given by Bade et al. (2006).

Copy Rights

3.2 Adaptive Correction

The quality of laser scanned 3D point clouds is mainly determined by the direction of projection and the viewing direction of the camera onto the object's surface. This fact can be exploited in order to adaptively remove redundant information. After registering the point clouds from different scan operations and sensors, the same small neighborhood region N often has been multiply sampled and contains sample points in different quality (see Figure 4). Points of lower quality downgrade the influence of points with high quality when applying neighborhood-based operations to these regions. Therefore, low quality points should be removed from the merged point set.

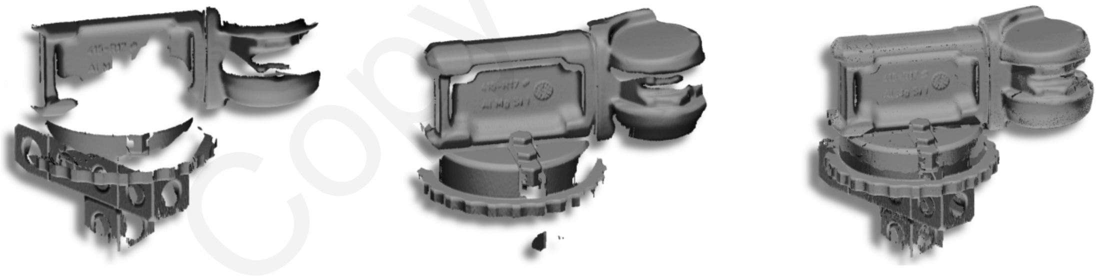
To minimize the number of points that are removed, the minimal neighborhood radius r_{min} should be selected for merging. Useful definitions for $r_{min}$are either based on the distance between points or two scanlines Δ_s (Eq. (2)) or the expected uncertainty of the 3D scanner σ_M. For the analyzed point sets, Δ_s is larger than the measuring uncertainty in most cases, which causes relatively large neighborhood, and thus the removal of too many points. The uncertainty (0.1mm) is more suitable for this purpose, but may be too small if Δ_s is large. A further adaptive measure is given by determining the typical (average) distance of two points on the considered scanline Δ_p. Since the directions of all three measures are different, they are interpreted as a vector whose length is the radius r_c for the neighborhood in which the correction is performed.

$$r_c = \sqrt{\Delta_p^2 + \frac{\Delta_s^2}{4} + \sigma_M^2} \qquad (11)$$

For the normalized quality values q_i of all points p_i in the resulting neighborhood $N(r_c)$, the average $\overline{m}_q$ is computed. This value serves as a threshold which defines that a point p_i in $N(r_c)$ should be removed, if its quality is lower than the threshold $\overline{m}_q$. Specifically:

As a result, the redundancy from the regions N is avoided by removing low quality points (see Figure 4(c)). For the flexible laser without constant distances between the scanlines Δ_s is set to zero.

It was also noticed that an adaptive correction for points with low quality on the basis of neighboring high-quality points by weighting is not reasonable. On the one hand, the uneven sampling would still remain and on the other, the necessary low weight for the considered point causes only a weighted interpolation and smoothing between the neighbors with negligible influence of the point itself.



(a) (b) (c)

Figure 4. Quality-based merging of redundant surface parts from different scan operations and sensors. The data sets obtained from the lower and upper sensor with their corresponding viewing quality are given in (a) and (b). The merged result is shown in (c).

3.3 Adaptive Simplification

Multiple scanning of the object's surface is often necessary to assure the capturing of all interesting surface parts. This often results in very large point clouds, with no significant increase of information due to a higher density at overlaying regions. After having merged overlaying points, the point density usually is still very high. In order to increase the computation performance of the following algorithms, a point-based simplification is applied. Usually, there is a differentiation between uniform and adaptive non-uniform procedures. To ensure a constant, uniform distance d_u between neighboring points, all points in the neighborhood of a point p with $r = d_u$ are removed (see Figure 5 (a)). The advantage of this approach is the

Copy Rights

high computation speed, but its disadvantage is that it does not regard local surface properties. But especially this adaptivity allows to remove more points in planar regions, than at edges and in curved regions.

For visualization purposes, Pauly et al. (2002) presented an iterative method. They compute the local surface variation obtained from a covariance analysis of the k nearest neighbors. From the eigenvalues λ_i, derived from the eigenvectors of the covariance matrix, they determine the corresponding normal vectors and achieve a consistent orientation with the procedure described in Section 3.4. The surface variation σ for a point p is then defined by λ_0 as the deviation from the plane, spanned by the mass center of the neighborhood (with size n) and the normal vector.

$$\sigma_n(p) = \frac{\lambda_0}{\lambda_0 + \lambda_1 + \lambda_2} \qquad (12)$$

For example, a zero deviation $\sigma_n(p)$ indicates that all points in the neighborhood of p lie in a plane, and if all eigenvalues have the same length, i. e. $\sigma_n(p) = 1/3$, a completely isotropically point distribution can be assumed. Points that cause the smallest error, are removed from the set by an edge collapse operation in an iterative manner. When using point cloud simplification as preprocessing for surface measurements, the input is a desired distance value d_a between two neighboring points that is only allowed to decrease in strongly curved regions and at edges (see). The necessary local surface properties for the adaptivity of the approach is then given by the curvature measures κ, that have been efficiently derived from the scanline analysis (see Figure 5(b)). Therefore, for each point p of the point set P all neighbors in the neighborhood d_a are identified with the help of the kd-tree. For each p_i, its value $\bar{\kappa}_i$ from equation (7) is used to determine the linear scale s_r that describes how much points must remain in the neighborhood of p_i, whereas p_i itself is never removed:

$$s_r = N_p(d_a) \cdot \bar{\kappa}_i \qquad (13)$$

If s_r is always set to zero, all points except p_i are removed and the procedure equals a uniform simplification. Otherwise, points with low quality $\bar{q}_i$ are removed first. To ensure that edge points are not removed by processing planar neighborhoods, the procedure is applied to the edge points first. This is achieved by dividing the point set into two subsets containing significant edge points on the one hand, and all remaining points on the other hand. The computation performance is increased, since points are not deleted from the kd-tree but labeled as removed (also see DeCoro (2007)). Although the tree becomes unbalanced, the whole procedure is much faster than re-balancing the tree by removing a point. At the end of the procedure, the tree is simply restored by unlabeling the knots.

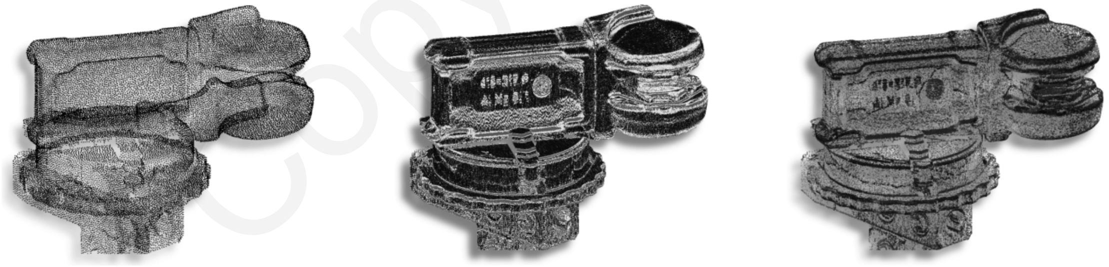


Figure 5. Point cloud simplification with r = 0.5. The result of the uniform simplification is shown in (a), the scanline curvature (b) is used for an adaptive simplification (c). For a better visualization, illustration (c) shows the resulting density in gray levels instead of single points.

4. CASE STUDIES

The proposed methods were applied to different models in order to evaluate their effectiveness. Figure 6 illustrates the models and the processing pipeline. The polygonal approximation of the initial point sets in the first column shows the influence of noise, redundancy and uneven sampling. The most problematic case is an overlay of noisy point sets, since the local surface is then represented by multiple point layers. This effect is significantly reduced by the correction step, which solves the redundancy by removing low quality points in local neighborhoods. The correction is additionally supported by a following smoothing procedure, which is adapted to the scanline curvature (second column) to preserve edges while smoothing noise. Since the smoothing performs a weighted averaging, an uneven sampling is implicitly corrected, too. In order to increase the computation speed of the following local point processing operations, the number of points has been adaptively reduced. The models also show different kinds of curvature of small and large areas with sharp and more smooth edges. Since the simplification procedure is also based on this curvature information, it reduces the point number depending on the strength of the curvature (third column). The point clouds processed this way show a significantly increased quality of their polygonal approximation (last column), although the point density is also significantly reduced.

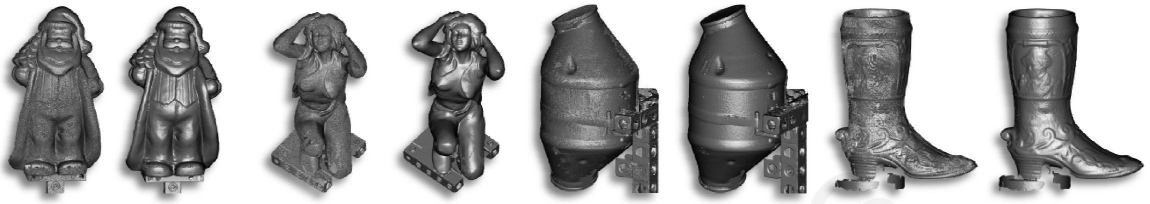


Figure 6. Case studies for the proposed methods at the examples of different point clouds.

In order to evaluate the efficiency of the proposed approaches Table 1 shows the total computation timings. For data sets up to 1 million points the algorithms take only one to two seconds. For larger sets memory access becomes an additional significant issue.

Table 1. Performance evaluation for the proposed methods at the example of the models in Figure 6 using an Intel Core2Duo 2.53GHz, 4GB RAM.

model number	number of data points	smoothing with radius 0.1mm	correction with radius 0.1mm	simplification to 100.000 pts
1	12.757.244	37.7 sec.	37.7 sec.	13.1 sec.
2	6.201.172	12.9 sec.	12.9 sec.	6.7 sec.
3	1.488.444	2.6 sec.	2.6 sec.	1.6 sec.
4	647.590	0.9 sec.	0.9 sec.	0.8 sec.

Copyrights

5. CONCLUSION

We presented strategies for the management, analysis and processing of point clouds derived from 3D scanners. The methods are fast and robust and adaptively derive their parameters from the data in most cases. The employed kd-tree only stores pointers to the complex scan data structures, which is memory efficient and provides immediate access to the additional system information generated during the scan process. A further improvement to the query time is provided by range-trees, which enable queries in $O(\log^3 n + p_r)$ time.

The smoothing and point-based simplification can significantly increase the performance when creating polygonal approximations of large data sets. Because on the one hand, the number of points to be processed is reduced, and on the other hand the smoothing operations reduce topological distortions due to noise. The presented methods also benefit from utilizing the given system information, like quality measures, uncertainty estimations, and scanline distances to increase the degree of automation and their adaptivity.

REFERENCES

Ahn, S.J. (2004): *Least Squares Orthogonal Distance Fitting of Curves and Surfaces in Space*. Lecture Notes in Computer Science. Springer, Berlin.

Attene, M.and Spagnuolo, M. (2000): Automatic surface reconstruction from point sets in space. *Computer Graphics Forum 19*, pp. 457–465.

Bade, R. et al. (2007): Reducing artifacts in surface meshes extracted from binary volumes. *Journal of WSCG* 15, 67–74

Bade, R. et al (2006): Comparison of fundamental mesh smoothing algorithms for medical surface models. In: *Simulation und Visualisierung*, SCS-Verlag, pp. 289–304.

Chen, Z. and Chou, H.L. (2006): New efficient octree construction from multiple object silhouettes with construction quality control. In: *18th Int. Conf. on Pattern Recognition (ICPR 2006)*. Vol. 1. pp. 127–130

de Berg M. et. al. (2008): *Computational Geometry: Algorithms and Applications*. 3rd ed. Springer

DeCoro, C. and Tatarchuk, N. (2007): Real-time mesh simplification using the GPU, In: Proc. Interactive 3D graphics and games I3D'07, pp. 161–166, ACM.

Dey, T.K. et al, (2005): Normal estimation for point clouds : A comparison study for a voronoi based method. In: *Eurographics Sympos. on Point-Based Graphics*. pp. 39–46

Gois, J.P.; et al. (2007): Robust and Adaptive Surface Reconstruction using Partition of Unity Implicits. Computer Graphics and Image Processing, SIBGRAPI 2007. pp. 95–104

Kriegel, H.-P. et al (2009): LoOP: local outlier probabilities., in Cheung David Wai-Lok et al. ed., 'CIKM' , ACM, , pp. 1649–1652.

Lange, C.and Polthier, K. (2005): Anisotropic smoothing of point sets. *Comput. Aided Geom. Des*. 22, 680–692

Miklos B. et al. (2010): Discrete Scale Axis Representations for 3D Geometry. In *ACM Transactions on Graphic. Vol* 29(4), pp. 1–10.

Mitra, N.J. et al. (2004): Estimating surface normals in noisy point cloud data. Special issue of *Int. J. Computational Geometry and its Applications* 14, pp. 261–276

Nagai Y. et. al, (2009): Smoothing of Partition of Unity Implicit Surfaces for Noise Robust Surface Reconstruction. Comput. Graph. Forum 28(5), pp. 1339–1348

Nealen, A. et al, (2006): Laplacian mesh optimization. In: *ACM GRAPHITE*. Pp. 381–389

Nielson, G.M. (2004): Radial hermite operators for scattered point cloud data with normal vectors and applications to implicitizing polygon mesh surfaces for generalized csg operations and smoothing. In: *VIS '04: Proc. of the Conf. on Visualization '04,* Washington, DC, USA, IEEE Computer Society, pp. 203–210

Ohtake, Y. et al, (2003): Multi-level partition of unity implicits. In: *SIGGRAPH '03: ACM SIGGRAPH 2003 Papers*, New York, NY, USA, ACM Press, pp. 463–470

Park, J.C. et al (2006): Elliptic Gabriel graph for finding neighbors in a point set and its application to normal vector estimation, Computer Aided Design 38, pp. 619–626

Pauly, M. et al. (2003): Shape modeling with point-sampled geometry. In: *SIGGRAPH 2003*, ACM Press, pp 641–650

Pauly, M. et al, (2002): Efficient simplification of point-sampled surfaces. In: *VIS '02: Proceedings of the conference on Visualization '02*, Washington, DC, USA, IEEE Computer Society, pp. 163–17.

Press, W.H. et al. (2007): *Numerical Recipes in C++: The Art of Scientific Computing*. 3rd ed. Cambridge University Press, Cambridge, UK.

Teutsch, C., et al. (2007): Adaptive Real-Time Grid Generation from 3D Line Scans for fast Visualization and Data Evaluation., in IV07, IEEE Computer Society, pp. 177–184 .

Copy Rights

SLOTS – A MODELING LANGUAGE FOR SCHEDULING PROBLEMS

Thomas Scheidl*, Günther Blaschek*, Peter Feigl** and Norbert Lebersorger**
**Institute of System Software, Johannes Kepler University, Altenberger Str. 52, 4040 Linz, Austria*
***Institute of Business Informatics, Johannes Kepler University, Altenberger Str. 69, 4040 Linz, Austria*

ABSTRACT

In this paper, we describe SLOTS, a domain-specific modeling language for specifying scheduling problems. SLOTS allows the modeling of scheduling problems by defining, extending and redefining classes and attributes. A SLOTS specification not only describes the general structure of the problem, but also defines the format of the input data for particular problem instances.

KEYWORDS

Optimization, scheduling problems.

Copy Rights

1. INTRODUCTION

Scheduling problems are combinatorial optimization problems. They can be described as having some tasks or jobs that must be scheduled on resources so that given constraints are met. Resources are everything that is needed to fulfill a specific task. Examples are machines, raw materials and manpower. The jobs can be parts of a manufacturing process, activities etc. Jobs have a set of properties, e.g. they need a specific amount of a resource (e.g. 50 kW of energy), a deadline until the job must be completed, or a processing time. Furthermore, jobs can be related to each other. This means for instance that a job can only start after another job has completed. The goal in scheduling problems is to find optimal arrangements of jobs on the resources on a timeline (these arrangements are called "schedules") that meet all constraints (Blazewic et al., 2001). The function for expressing the quality of a schedule is called objective function and is either minimized or maximized. Examples are to minimize the latest completion date of all jobs or to maximize the earnings in a manufacturing process.

The goal is to automatically generate an optimizer that heuristically solves a scheduling problem. For this purpose, it is necessary to specify and describe scheduling problems in a general way. It must be considered that scheduling problems not only consist of the description of the structure of the system, e.g. having jobs with a specific processing time, but also have external data. So, a particular problem consists of a global part that describes the structure (the problem class) and a local part with the concrete data for a specific problem instance. This data can be the number of jobs to be optimized or the processing times of specific jobs.

To specify scheduling problems in such a general way, we developed SLOTS (abbreviation for Scheduling Language for OptleTs Schedulers). SLOTS is a declarative and object-oriented modeling language for describing scheduling problems. It allows to specify both the structure of the problem class and the data.

As the name of the language suggests, it is designed for generating optimizers based on the OptLets optimization framework (Breitschopf et al., 2005; Breitschopf et al., 2006). This is a general evolutionary framework that starts with a simple, non-optimal solution and incrementally tries to improve this solution by means of so-called OptLets. An OptLet is an algorithm that receives an existing solution, performs some typically simple operations on it, resulting in a new solution. The framework manages the solutions and the OptLets and returns the best solution at the end of an optimization run. In addition to the SLOTS language itself, we developed a compiler that generates the necessary C++ code for an optimizer based on the OptLets framework. Our goal was to generate OptLet optimizers for different classes of scheduling problems just by specifying the problem in SLOTS, running the SLOTS compiler and building the resulting C++ code.

In section 2, we refer to comparable languages in the field of scheduling problems and describe the idea of declarative programming languages in combination with object orientation. Section 3 describes the basic language concepts of SLOTS: attributes and classes. In section 4, we show the connection to the input data. Section 5 presents the scheduling-specific concepts in SLOTS. Finally, in section 6, we present results and summarize our experiences.

2. RELATED WORK

SLOTS is a language that unites declarative and object-oriented approaches in order to succinctly describe scheduling problems. This means it can be compared with languages for modeling optimization problems, but also with languages that combine declarative and object-oriented programming.

We have investigated a number of languages that can be used for modeling optimization problems, including scheduling problems. Most of them are algebraic modeling languages, such as OPL (Optimization Programming Language) (Hentenryck, 1999), GAMS (General Algebraic Modeling System) (Brooke et al., 1992), Mosel (Colombani and Heipcke, 2009), and AMPL (A Modeling Language for Mathematical Programming) (Fourer et al., 2002). Exceptions are Comet (Hentenryck and Michel, 2005), a constraint-based language with explicit support for scheduling problems, IF/Prolog (IFComputer, 2010), and three domain-specific languages, RCSpec (Zentner et al., 1998) and Vishnu (Montana et al., 2007).

Figure 1 summarizes the criteria we formulated. A general description language for scheduling problems should fulfill all of these. As the figure shows, none of the investigated languages satisfies the entire list of requirements. Our main critique of the existing description languages is that they are closely connected to the solver; intricate details about the solution technique must be known in order to accurately describe the problem. Our goal was a general modeling language that describes the problem, not the way to solve it.

	OPL	GAMS	Mosel	AMPL	Comet	IF/Prolog	RCSPec	Vishnu
Multiple objective functions	○	●	●	●	●	●	○	●
Separation of model and data	●	○	●	●	●	●	○	●
Explicit support for scheduling problems	●	○	○	○	●	○	●	●
Text-based specification language	●	●	●	●	●	●	●	○
Solver-independent problem description	●	●	●	●	○	●	●	●

Figure 1. Aspects in related languages

Copy Rights

3. BASIC CONCEPTS

SLOTS is a purely declarative language, i.e. it contains only declarative, but no imperative language elements. A problem specification written in SLOTS consists of a set of declarations of types (including classes), attributes (variables or constants) and objectives (the aim of the underlying optimization problem).

3.1 Attributes and Data Types

Attributes are declared by specifying a data type, a name and a value:

```
 integer amount = 5;
```

```
float costs = 3.65;
time t = 0.5;
```

The values of attributes declared in this way cannot change, i.e. they are actually constants. An alternative to assigning a constant value to an attribute is to declare it as input data:

```
time duration = <>;
```

The angle brackets <> mean that the value of the attribute should be read from an input file that is passed to the generated optimizer. The value is still immutable after it has been read from the input file. Attributes that shall be altered during the optimization (i.e. assigned a value by the optimizer) are declared by specifying a possible value range:

```
time startingDate within 0..1000;
```

These attributes can be seen as variables that can hold values within the specified range. However, it is not possible to assign values to these variables within the SLOTS language (as there is no assignment statement). The values of these attributes are modified only by the generated optimizer. Attributes can also be defined by means of arbitrary expressions, resulting in so-called computed attributes:

```
// constant
integer dblAmount = amount *2;
// immutable
float totalCosts = duration*costs;
// variable
time endDate = startingDate+duration;
```

Computed attributes can be constant (value can be determined at compile time), immutable (value can be determined after reading the input data) or variable (value may change during the optimization), depending on what other attributes are used for computing the value. If a computed attribute depends on an optimized attribute, its value changes whenever the value of the optimized attribute changes.

There are three numeric data types: integer, float and time (a fixed point data type with a configurable resolution, used for both points in time and durations). Furthermore, there is a boolean data type. Multiple values can be stored in a single attribute by using arrays:

```
integer [] values = {1, 2, 3};
float [] costs = <>;
time [10] times = <>;
```

The size of an array needs not to be specified as it can be determined from the initial value or when reading the input data. Specifying the size can be useful for requiring a particular number of elements in the input data. Arrays are always constant or immutable, i.e. they cannot be changed by the optimizer.

3.2 Classes and Objects

SLOTS is an object-oriented language, so classes and objects are important core concepts. Classes are introduced via type declarations and can be used for declaring multiple objects that share a set of properties:

```
type MyClass = class {
  integer a = 1;
  integer b = 2;
};
```

A class consists of a set of attributes declared with default values. When objects of a class are declared, explicit values may but need not be specified for the attributes. For attributes that are not specified in the declaration of an object, the default value specified in the class is used:

```
MyClass obj1; // a=1, b=2
MyClass obj2 = {a=3}; // a=3, b=2
MyClass obj3 = {a=3, b=4}; // a=3, b=4
```

The first object obj1 is declared without specifying a value which means that all attributes have the values specified in the class. For obj2 and obj3, the values are given in form of so-called object literals containing explicit values for one or both class attributes a and b. Like other object-oriented languages, SLOTS supports inheritance, i.e. deriving subclasses that override existing attributes and add new ones:

```
type ExtClass = class of MyClass {
  a = 3; // redefine attribute a
  integer c = 5; // add new attribute c
};
```

Copy Rights

SLOTS supports polymorphism, i.e. an attribute of class MyClass can hold a value of class ExtClass. For attributes read from the input data, this means that the input data may contain objects of the base class or a subclass. Whereas imperative object-oriented languages allow to override methods, SLOTS uses overriding of attributes (i.e. constants or variables). Overriding an attribute changes its default value. Accesses to class attributes are dynamically bound, as the following example shows:

```
type BaseClass = class {
  integer a = 1;
  integer b = 2;
  integer c = a+b;
};
type ExtClass = class of BaseClass {
  a = 3;
}
BaseClass baseObj; // c=3
ExtClass extObj; // c=5
```

In this example, the attribute c is a computed attribute with a value depending on the attributes a and b which are accessed using dynamic binding. Dynamic binding not only comes into play when an attribute is overridden in a subclass, it also becomes effective when attributes are declared with an explicit value in an object literal:

```
BaseClass baseObj2 = {a=2}; // c=4
ExtClass extObj = {b=4}; // c=7
```

Specifying an explicit value for an attribute within a literal has the same effect as overriding an attribute in a subclass. So, each object literal can actually be seen as a singleton of an anonymous class derived from the class specified in the declaration of the object. When overriding an attribute, it is also possible to change a constant value to a value read from the input data or vice versa or even to change an optimized value to a constant value or a value read from the input data or vice versa:

```
type BaseClass = class {
  integer a = 1;
  integer b = <>;
  integer c within 1..10;
};
type ExtClass = class of BaseClass {
  b = 3;
  c = <>;
}
```

For objects of type BaseClass, b is read from the input data and c is a variable between 1 and 10, set by the optimizer. For objects of type ExtClass, b is not read from the input data but set to a constant value of 3, and c is no longer variable but read from the input data (and immutable afterwards). In conventional object-oriented languages, overriding mainly affects method implementations. In SLOTS, overriding is used to modify values of attributes and to redefine the way how values of attributes are determined.

3.3 Objectives and Constraints

At the end of each SLOTS specification, an objective function must be specified. Objective functions can be either minimized or maximized:

```
minimize totalTime;
```

The specified expression must be an optimized attribute or be computed from at least one optimized attribute. The generated optimizer will try to find a combination for all optimized attributes such that the declared objective becomes minimal or maximal (whatever is specified). It is possible to define multiple objective functions. In this case, a multi-objective optimizer is generated which produces a pareto front instead of a single solution, i.e. a set of solutions that are not dominated by other solutions.

Scheduling problems (and optimization problems in general) often have constraints that must be satisfied. In SLOTS, constraints can be specified by declaring attributes of the special type constraint:

```
constraint meetDeadline = completionDate <= deadline;
```

The generated optimizer will respect all constraints that are contained in the global attribute constraints or in a class attribute constraints. Per default, these attributes are defined as empty arrays. In order to enable checking of a constraint, the constraints attribute must be overridden accordingly:

Copy Rights

```
constraints = {meetDeadline};
```

The optimizer will try to find a combination for all optimized attributes such that all constraints are satisfied.

4. CONNECTION TO INPUT DATA

SLOTS allows to define the values of attributes in external files. The same SLOTS specification can be used for multiple problem instances by changing only the input data. The input data file contains all specific data of a problem instance and the SLOTS specification defines the common structure of the problem.

4.1 Declaration of Input Data

As already mentioned, attributes can be declared to be read from an input file:

```
float costs = <>;
```

This declaration means that the float value costs is read from an input file. The input value is mandatory. In some cases, it is useful to define default values for attributes:

```
float costs = <=2.0>;
```

In this case, the default value of costs is set to 2.0. Attributes with a default value are optional in the input data. If the value is present in the input data, it overrides the default value. Sometimes, it is convenient to use a different attribute name in the input data than in the SLOTS specification, e.g. if the names in existing input data come from a legacy system and differ from the names used in the SLOTS specification. Therefore, SLOTS allows to specify different names for input data attributes:

```
float costs = <price>;
float penalty = <weight=0>;
```

Here, the first input data attribute has the name “price”. SLOTS connects the input attribute price to the corresponding SLOTS attribute costs. It is also possible to combine attribute renaming and default values, as the declaration of the attribute penalty (with the input name weight and a default value of 0) shows.

4.2 Input Data Format

SLOTS uses XML as its default input data format, because it is a generally accepted standard. The structure of the input file is specified by the structure of the SLOTS specification. The root container which contains the attributes in the global scope is called “slots”. This node contains all elements marked as to be read from input in the SLOTS specification. The following example shows a global SLOTS attribute costs of type float.

```
float costs = <price>;
```

Input data example:

```
<slots >
...
<price type="float">3.0</price>
...
</slots>
```

The name of the input data attribute is redefined to price. The name of an attribute in the SLOTS specification defines the name of the attribute in the input data. For better readability of the input data, each XML element has an optional XML attribute type which describes the content of the XML element. In the example, the type of price is float.

5. SCHEDULING-SPECIFIC CONCEPTS

SLOTS is a language designed to express a wide range of problems, but the main focus is on scheduling problems. A number of predefined classes and attributes makes the definition of scheduling problems straightforward in most cases.

Copy Rights

5.1 Scheduling Problems

The two basic abstractions in scheduling problems are resources and jobs. A job (also known as task or operation) models an activity that is executed on a number of resources over time. A resource is any item or commodity that is required by a job to ensure that it can be executed successfully. Painting a car door red is a job that requires red paint, a brush, a car door, and a painter. Figure 2 shows how nine jobs could be run on three resources.

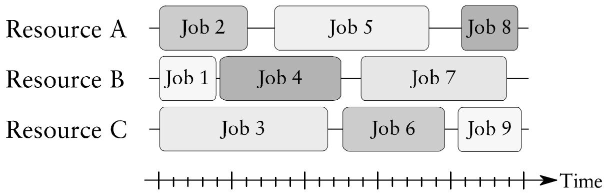


Figure 2. Scheduling problem

The two main classes that SLOTS provides to support scheduling problems are Job and Resource. Job contains attributes that represent due dates, release dates, deadlines, precedence, and other concepts. Resource models any renewable or non-renewable resource that is needed to run a job. The predefined attributes of Job and Resource have default values that are suitable for many cases, but can be redefined in order to describe more sophisticated scheduling problems. The predefined attributes help to keep descriptions of simple cases short but nevertheless allow the formulation of rather complex scenarios by overriding them in subclasses.

5.2 Example: Common Due Date

The following example shows how to formulate a concrete scheduling problem in SLOTS: the Common Due Date problem (Brucker, 2004). Each job has a penalty that is computed as its weighted earliness or tardiness, relative to a common due date for all jobs. The objective is to minimize the sum of all jobs' penalties.

```
time commonDueDate = <>;
type WeightedJob = class of Job {
  processingTime = <>;
  dueDate = commonDueDate;
  integer earlyWeight = <>;
  integer tardyWeight = <>;
  integer penalty = earliness * earlyWeight + tardiness * tardyWeight;
};
```

The attribute commonDueDate is declared at the top level outside the definition of the job class. This is a so-called global attribute that is independent of jobs. The value of commonDueDate will come from the input data. The class WeightedJob inherits from Job. The two predefined (and inherited) attributes processingTime and dueDate are defined with new values. The processingTime is read from the input data and thus overrides the default value of 1, and the dueDate is the same as the value of the global attribute commonDueDate. A new attribute penalty with type integer is introduced, its value is the sum of earliness times earlyWeight and tardiness times tardyWeight. Both earliness and tardiness are predefined attributes of class Job, and both earlyWeight and tardyWeight are user-defined job-specific attributes that get their values from the input data. Since a job cannot be early and tardy at the same time, the penalty is influenced by either earliness or tardiness. This succinctly specifies all data pertinent to the jobs, each job has its own processingTime, they all share the commonDueDate, and each job has its own penalty.

For this example, we do not need a custom Resource class, the default will do fine:

```
Resource resource;
resources = {resource};
WeightedJob [] weightedJobs = <>;
jobs = weightedJobs;
```

One resource with the name resource is defined, and the predefined global array resources is redefined as that single resource. Also a global array weightedJobs is declared as initialized from the input data,

Copy Rights

containing WeightedJobs. The predefined global array jobs is then set to the value of weightedJobs. These two arrays, resources and jobs, define all resources and jobs for the SLOTS optimizer, which then tries to find a good assignment of jobs to resources, where the quality of an assignment is defined by the objective function:

```
 minimize sum(penalty) of weightedJobs;
```

These 11 lines fully specify the problem, and are sufficient to generate an optimizer. This is mainly due to the fact that SLOTS already contains a dozen predefined classes and more than 50 predefined attributes that are specifically tailored towards scheduling problems. There is no need to explicitly declare what the earliness or tardiness of a job is, SLOTS already knows how these can be calculated from the due date and completion date. Some predefined attributes introduce implicit constraints: by declaring a release date, a job's possible position within a schedule is constrained. The predefined attributes allow to express mutual exclusion (by declaring a set of incompatible jobs), required resources for running a job, precedence relationships (including time lags), deadlines, job hierarchies (where jobs are composed of smaller operations), etc. A number of attributes are automatically computed, such as the completion date, the lateness, the earliness, and the tardiness of a job. A resource has predefined attributes for dealing with machine speed, renewability, maximum capacity, times during which the resource is unavailable, setup times, and more.

Any properties that are not expressible via predefined attributes can be added as custom attributes or constraints (as described in section 3.3).

6. CONCLUSION

The SLOTS language allows the specification of scheduling problems in a compact way. It has been successfully used for a set of different scheduling problems, including well-known problems such as Job Shop, Open Shop, Flow Shop or Project Scheduling. In all cases, the problem specification of the complete problem is about 20 lines of SLOTS code. Using XML as the input and output format brings the advantage of having a standardized format. The input format is defined so that SLOTS developers can easily connect a SLOTS specification to the input and the output file formats.

The SLOTS compiler for the OptLets framework creates fully functional problem-specific optimizers. Table 1 shows a summary of different case studies using the SLOTS compiler. For each of the problem classes, the table presents the lines of code for the SLOTS specification, the generated lines of code for the OptLets optimizer and for 6 randomly selected problem instances with different sizes the average deviation to known optima.

Table 1. SLOTS Compiler: Case studies

Problem	SLOTS LoC	Generated LoC	Deviation to Optima (Avg.)
Common Due Date (Beasley, 2010)	13	5553	8.5%
Project Scheduling (TU Munich, 2010)	22	10401	7.5%
JobShop (Beasley, 2010)	19	7609	24.2%
OpenShop (Beasley, 2010)	19	7206	1.6%
FlowShop (Beasley, 2010)	19	7609	19.1%

These results show that the generated optimizers deliver acceptable but not yet optimal results in many cases. Nevertheless, the generated optimizer is a good starting point and can be extended in a variety of ways. In particular, new OptLets can easily be added in order to improve the result of the optimization.

The combination of object-orientation and declarative programming has proven to be very useful for formulating scheduling problems. Especially the predefined classes in SLOTS such as jobs and resources enable a rapid and compact specification of the problem. For many problem classes, just some of the predefined attributes need to be changed. Every redefinition of a built-in attribute changes the problem attributes detected by the SLOTS compiler, which leads to a different optimizer. Inheritance allows to easily extend the predefined classes for adding new problem-specific attributes.

SLOTS satisfies all the criteria we postulated in section 2 and is thus a general specification language better suited for describing scheduling problems than any of the investigated languages.

Copyrights

In this paper, we presented the basic concepts of the SLOTS modeling language and showed how to formulate scheduling problems by using and extending the predefined classes and attributes in an easy way. Our experiments with different problems showed that SLOTS is suitable for a broad range of scheduling problems

REFERENCES

Åkesson, J., Gäfvert, M., and Tummescheit, H., 2009. Jmodelica – an open source platform for optimization of modelica models. *Proceedings of MATHMOD 2009 – 6th Vienna International Conference on Mathematical Modelling*.

Beasley, J. E. (2010). OR-Library. *http://people. brunel.ac.uk/~mastjjb/jeb/info.html*.

Blazewic, J., Ekcer, K., Pesch, E., Schmidt, G., and Weglarz, J., 2001. *Scheduling Computer and Manufacturing Processes*. Springer, second edition.

Breitschopf, C., Blaschek, G., and Scheidl, T., 2005. Optlets: A generic framework for solving arbitrary optimization problems. *WSEAS Transactions on Information Science and Applications, 2(5)*.

Breitschopf, C., Blaschek, G., and Scheidl, T., 2006. A comparison of operator selection strategies in evolutionary optimization. *Proceedings of the 2006 IEEE IRI International Conference on Information Reuse and Integration*.

Brooke, A., Kendrick, D., and Meeraus, A., 1992. *GAMS: A User's Guide*. The Scientific Press.

Brucker, P. (2004). *Scheduling Algorithms*. Springer.

Colombani, Y. and Heipcke, S., 2009. Mosel: An Overview. *http://www.dashoptimization. com/home/downloads/pdf/mosel.pdf*.

Fourer, R., Gay, D., and Kernighan, B., 2002. *AMPL: A Modeling Language for Mathematical Programming*. Duxbury Press.

Hentenryck, P. V., 1999. *The OPL Optimization Programming Language*. MIT Press.

Hentenryck, P. V. and Michel, L., 2005. *Constraint- Based Local Search*. MIT Press.

IFComputer, 2010. IFProlog. *http://www. ifcomputer.co.jp/IFProlog/home_en.html*.

Montana, D., Hussain, T., and Vidaver, G., 2007. A genetic-algorithm-based reconfigurable scheduler. *In Dahal, K., Tan, K., and Cowling, P., editors, Evolutionary Scheduling*.

TU Munich, 2010. Project Scheduling Problem Library – PSPLIB. *http://129.187.106.231/psplib/*.

Zentner, M., Elkamal, A., Pekny, J., and Reklaitis, G., 1998. *A language for describing process scheduling problems*. Computers and Chemical Engineering, 22(1):125–145.

Copy Rights

ACCELERATE TWO-DIMENSIONAL CONTINUOUS DYNAMIC PROGRAMMING BY MEMORY REDUCTION AND PARALLEL PROCESSING

Yukihiro Yoshida, Koushi Yamaguchi, Yuichi Yaguchi, Yuichi Okuyama,
Ken-ichi Kuroda and Ryuichi Oka
The University of Aizu
Aizu-wakamatsu, Fukushima, Japan

ABSTRACT

This paper contains a proposal for optimizing and accelerating the computation of two-dimensional continuous dynamic programming (2DCDP). 2DCDP processing is optimized by memory reduction and parallelization using OpenMP. We apply buffer resizing and utilize toggle-type buffers to reduce the required memory size. In addition, same-rank processes and pixel correspondence calculation are parallelized by OpenMP instructions to reduce the computation cost/time of 2DCDP. For accumulation, we also apply a realignment of buffering addresses for SIMD on multi-cores/multi-processors. The experimental results show that the computational time and the memory usage have reduced to about 1/4 and 1/5 of the original ones, respectively. Moreover, the concurrency of 2DCDP hot-spot is improved from 5.8 to 7.1 on a quad-core CPU with 8 threads.

KEYWORDS

Image recognition, Pixel correspondence, Continuous DP, Parallelization

Copy Rights

1. INTRODUCTION

With the development of high-frequency, high-density, and multi-cored central processing units (CPUs) or graphic processing units (GPUs), developers can implement complex algorithms for high-efficiency applications. The fields of image recognition, especially, need many complex algorithms for image retrieval, segmentation, matching, stereography (Ohtaet. Y et al., 1985, Okutomi. M et al., 1993), or 3D object construction (Xiaojun. Z et al., 2002, Iseki. K et al., 2008). Because of the importance of real-time processing for image recognition, high-performance processors with process optimization are required. In fact, most image recognition algorithms such as the scale-invariant feature tracker (SIFT) (Lowe. D, 2004), histograms of oriented gradients (HoG) (Dalal. N et al., 2005), and template or block matching (TM, or BM) (Pereira. S et al., 2000, Zhu. S et al., 2000), are already implemented on multi-cored CPUs or GPUs.

Two-dimensional continuous dynamic programing (2DCDP) (Yaguchi. Y et al., 2008) is well known as a pixel-matching algorithm applicable to non-linearly deformed images. It was proposed as a technique for optimum matching of pixel-wise matching. 2DCDP can acquire the pixel correspondence between an input image and a reference image with very high accuracy even for affine transformation and non-linear segmentation relative to other existing techniques. 2DCDP algorithm can very effectively use a pixel-correspondent relation for not only image recognition but also other applications, such as segmentation, pixel flows of image correspondence (Kawashima. Y et al., 2009), and three-dimensional shape reconstruction. Image recognition and segmentation can be simultaneously processed by full-pixel correspondence. However, it requires enormous powerful machine, memory space, and complex calculation of O(N^4).

In this paper, we show an approach to optimize and accelerate the computation of 2DCDP. Its applications can benefit from solving the issues of big computational complexity. The 2DCDP processing requires enormous memory space for buffering four-dimensional results. We apply buffer resizing utilize of toggle-type buffers 2DCDP computation to reduce the memory requirement. In addition, parallelization by

OpenMP instructions and realignment of the buffering address for SIMD on multi-cores/multiprocessors are applied for acceleration.

2. TWO-DIMENSIONAL CDP (2DCDP)

2.1 Basic Concepts

2DCDP was proposed by Oka, et al. as an extension of continuous DP (CDP) to two dimensions among the spotting recognition methods in 1997 (Oka. R et al., 1997). Image-spotting recognition is a technique for the simultaneous segmentation recognition of an object through the pixel correspondence of images, as shown in Figure 1.

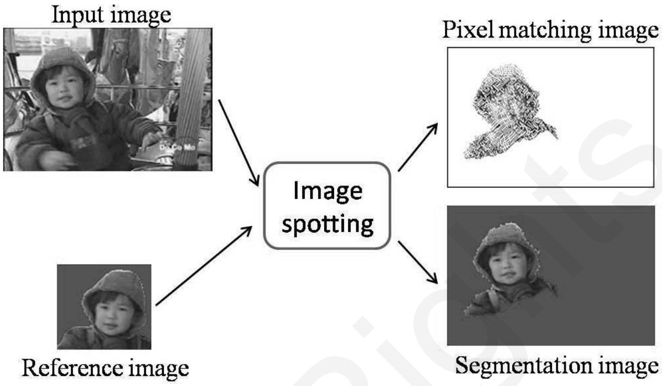


Figure 1. Image spotting

Copy Rights

2.2 The Algorithm of 2DCDP

2DCDP processing flow is shown in Figure 3 (a). The coordinates of input image *S* and reference image *R* are defined as follows:

$$S \triangleq \{\, (i,j) \,|\, 1 \le i \le I,\, 1 \le j \le J \,\}$$
$$R \triangleq \{\, (m,n) \,|\, 1 \le m \le M,\, 1 \le n \le N \,\} \qquad (1)$$

Next, mapping $R \rightarrow S$ is defined. In other words, ξandη, as used in the following equation.

$$(m,n) \in R \Rightarrow (\, \xi(m,n), \eta(m,n) \,) \in S \qquad (2)$$

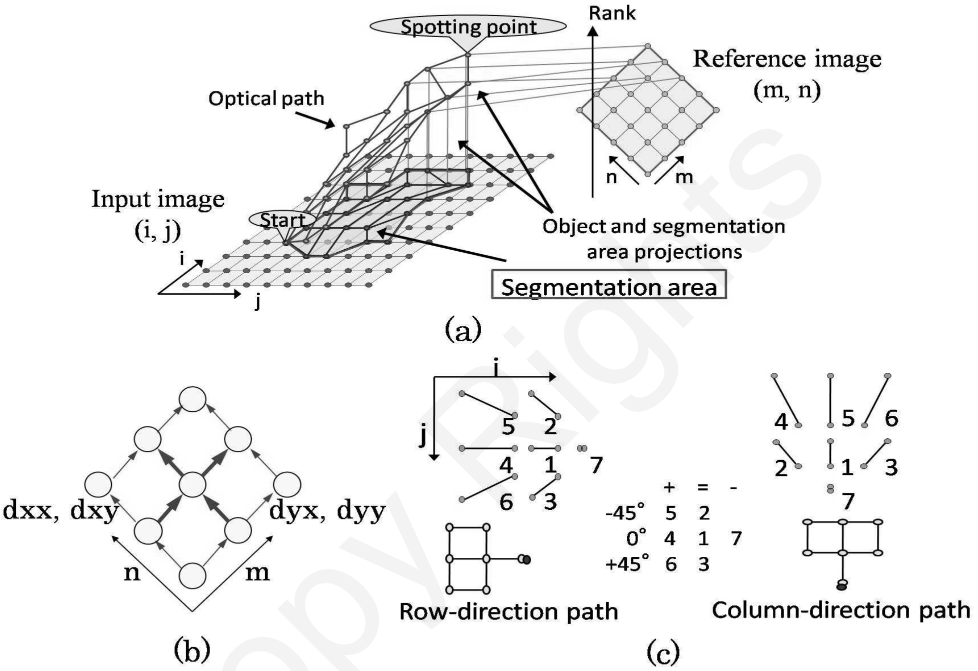
Then, the surface of accumulated local minimum $D(\hat{\ },i, \hat{\ },j, m, n)$ which $\hat{\ },i = \xi(M,N)$; $\hat{\ },j = \eta(M,N)$, or the accumulation cost from *R* to *S*, is defined as follows:

$$D(\hat{i},\hat{j},m,n) = \frac{1}{W} \min_{\zeta,\eta} \left\{ \sum_{m=1}^{M} \sum_{n=1}^{N} \right.$$
$$\left. \omega((m,n),\eta(m,n),m,n) d(\xi(m,n),\eta(m,n),m,n) \right\} \qquad (3)$$

The accumulation is processed to a diagonal direction from a starting point on the input image, as shown in Figure 2 (a). This processing axis is defined as a rank direction. The weight of the accumulation path is shown by *w*(*i, j, m, n*). The total weight of the best accumulation is shown by *W*. The local cost *d*(*i, j, m, n*) in the pixel correspondence when the pixel value of point (*i, j*) of input image S is defined as $S_p(i, j)$ and the pixel value of point (*m, n*) of reference image R is defined as $R_p(m, n)$ is defined as follows:

$$d(i, j, m, n) = \|R_p(m, n) - S_p(i.j)\| \qquad (4)$$

The accumulation is continuously processed from all starting points on the input image with the last point with the minimal accumulative cost is found. Point is defined as a spotting point. Figure 2 shows overview of 2DCDP and an example of mesh structure in 3D representation of 4D-working area (Figure 2 (a)). To smoothly cumulative calculate, the 2DCDP utilized four-tuple calculation techniques with binominal and diagonal direction and propagation (Figure 2 (b)). During accumulative calculation process, to support local rotation and local scaling, as Figure 2 (c), it is necessary to select 7 paths for calculation of two lower-rank nodes. After accumulative calculation, spotting point with accumulated local minimum value, which expressed in equation (3), then back-trace processing can reconstruct the segmentation space.



Copy Rights

Figure 2. Accumulation plot (a); Four-tuple calculation technique (b); and Local rotation and local enlargement (c)

3. ANALYSIS OF 2DCDP

3.1 Profiling

First, we survey performance of the 2DCDP algorithm which implemented by Yaguchi, et al. (Yaguchi. Y et al., 2008). This processing flow is shown as Figure 3 (a). For this survey, experimental machine specifications as is following: OS, Windows XP 64bit; CPU, Intel Core i7, 2.67GHz; and memory, 9GB SDRAM. Moreover, for the measurement of 2DCDP concurrency and memory usage, we use Microsoft Visual Studio 2008 Professional, Intel Parallel Studio evaluation edition (http://www.xlsoft.com/jp/products/intel/parallel/), and a CRN Monitor (http://www.runread.com/). Conventional 2DCDP is partial parallelized by OpenMP. As the profiling result of conventional 2DCDP, the ratio of each processing is as follows: calculation of accumulation cost is 87%; calculation of local cost is 6.4%; buffer initialization is 6.5%; sorting accumulation cost is 0.01%; and back-trace is 0.06%;. The execution costs are shown in Table 1. Images used in profiling are shown in Figure 4. The unit of image is a pixel. Clearly, there was a particularly large calculation time in accumulation processing, and the execution time increases rapidly as the image size increases.

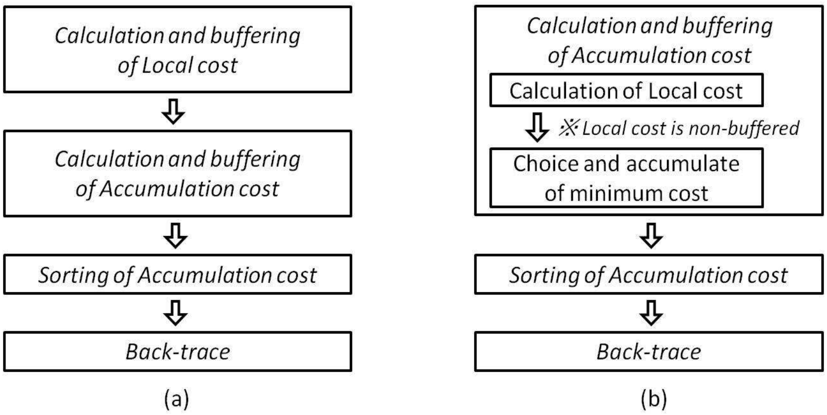

Calculation and buffering of Local cost
Calculation and buffering of Accumulation cost
Sorting of Accumulation cost
Back-trace
(a)
Calculation and buffering of Accumulation cost
Calculation of Local cost
※ Local cost is non-buffered
Choice and accumulate of minimum cost
Sorting of Accumulation cost
Back-trace
(b)

Figure 3. Conventional 2DCDP Flow (a); Proposed 2DCDP Flow (b)

Table 1. Execution results of conventional 2DCDP

Input image	Reference image	Computing time	Utilized memory
120 x 79	34 x 48	0.84 sec.	348 MB
120 x 79	53 x 71	1.92 sec.	796 MB
200 x 136	43 x 52	3.54 sec.	1350 MB
200 x 136	92 x 84	Inexecutable	

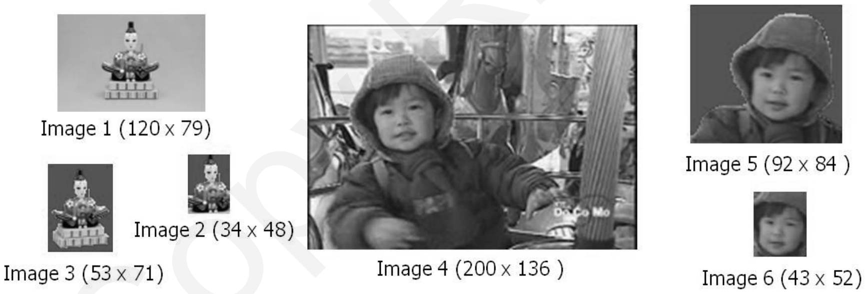


Figure 4. Images used in profiling

Copy Rights

3.2 Problems

The problems of 2DCDP processing are that it has time complexity of $O(N^4)$ to calculate the pixel correspondence of two-dimensional data, and the space complexity of $O(N^4)$ also requires memory storage as the result. Details of the methods to improve performance are explained in the next section.

4. IMPLEMENTATION OF 2DCDP

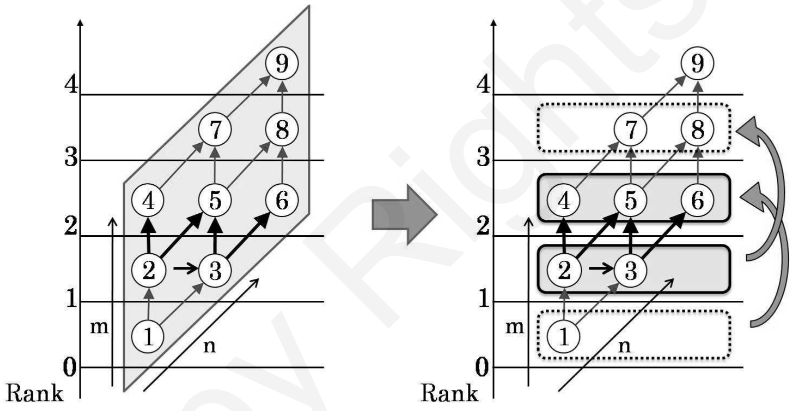
To solve the issues mentioned above, we apply computational methodology optimization. Various approaches for optimization of computational methodology, such as buffer resizing and the use of a toggle-type buffer, reduction and relocation of branches, improvement of multi-threading performance by parallely instruction, pipeline processing, and memory management optimization by function pointer, have already been proposed for other areas (Franchetti. F et al., 2009, Kim. H et al., 2009) but not intensively applied to 2DCDP.

In this paper, we apply some of the above methods, such as buffer resizing, using a toggle-type buffer, reduction and relocation of a branch, parallelization by OpenMP, and realignment of the buffering address for Single Instruction/Multiple Data (SIMD) to 2DCDP

4.1 Buffer Reduction

In the 2DCDP processing, all of the local costs are buffered after being calculated at the beginning because the local cost is calculated separately for two stages in the initial algorithm.However, the pixel correspondence from R_p to S_p is always constant through the current local path. Therefore, buffer utilization is reduced by calculating the local cost in each time when requested in accumulation processing. A new 2DCDP processing flow is shown in Figure 3 (b).

Next, all of the accumulation data is buffered in the 2DCDP because, in the initial algorithm, these four kinds of data are intended to be used in the back-trace processing. However, the current back-trace processing can be calculated using only the sum of four variables. Therefore, we use two toggle-type arrays, as shown in Figure 5. To obtain the accumulation cost, only two foremost ranks necessary for the accumulation are buffered. In other parts, four variables are gathered and buffered.



Copy Rights

Figure 5. Toggle-type buffering

4.2 Parallelization

The accumulation processing is parallelized using OpenMP in the same rank in Figure 6 (a) because processes in the same rank do not depend on each other. Moreover, because the corresponding point is constant at each time, the calculation of the local cost from the pixel correspondence and accumulation processing is parallelized using OpenMP, as shown in Figure 6 (b)

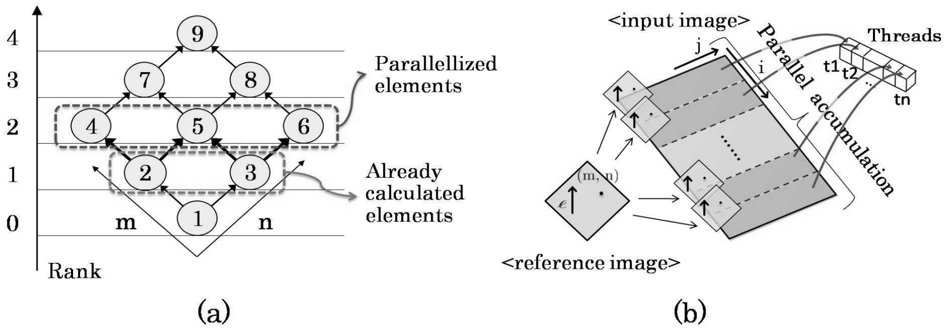


Figure 6. Parallelization of same rank processing (a); Parallelization of local cost calculation

4.3 Sorting of Memory Alignment

In the 2DCDP, the accumulation cost buffer is composed of a two-dimensional array. The accumulation process requires synchronization in a diagonal rank direction. Single Instruction/Multiple Data (SIMD) is a technique of simultaneous computation for the data on the adjacent memory address. Therefore, proposed 2DCDP processing can be applied to SIMD by realigning the sequence data so that the processing address in the rank is adjacent, as in Figure 7. We do not implement SIMD in this research, but the Intel C++ compiler supports automatic vectorization for SIMD. As a result, the processing time is accelerated.

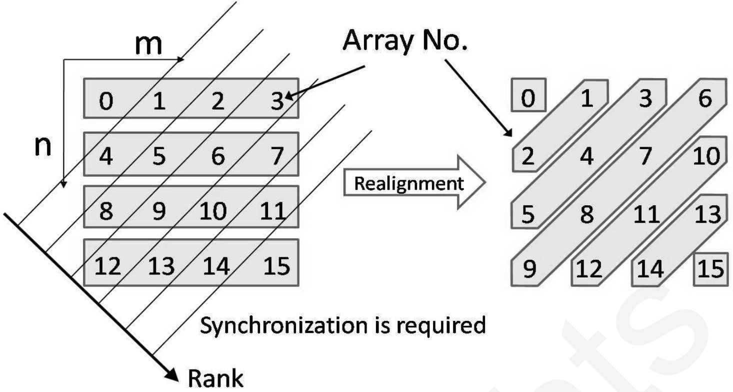


Figure 7. Realignment of accumulation buffer

4.4 Branch Reduction

In conventional 2DCDP, the processing of image boundaries is executed by four functions using a branch. Therefore, padding is implemented in the accumulation buffer to reduce the branch in Figure 8. In addition, the branch in the loop processing that does not depend on the loop variable is relocated out of the loop.

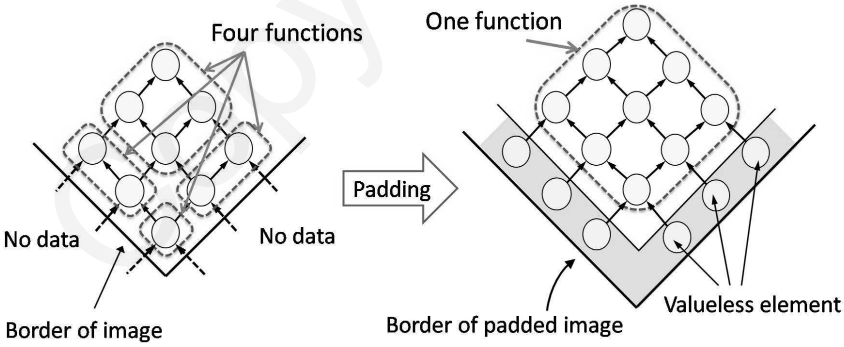


Figure 8. Padding of image

5. EVALUATION

5.1 Experimental Methods

We apply the proposed 2DCDP processing to several image combinations. The experiment environment is the same as profiling. The calculation time is measured by a timer library embedded to C++. The memory cost is measured by the CRN Monitor.

5.2 Experimental Results

The experimental results comparison of conventional 2DCDP and proposed 2DCDP is shown in Figure 9. Because of our implementation, the calculation time is improved to about 4.0 times of the original time, and the memory cost becomes 21% of the original memory cost. Moreover, the concurrency improves from 5.8 to 7.1 on the quad-core CPU calculable with 8 threads.

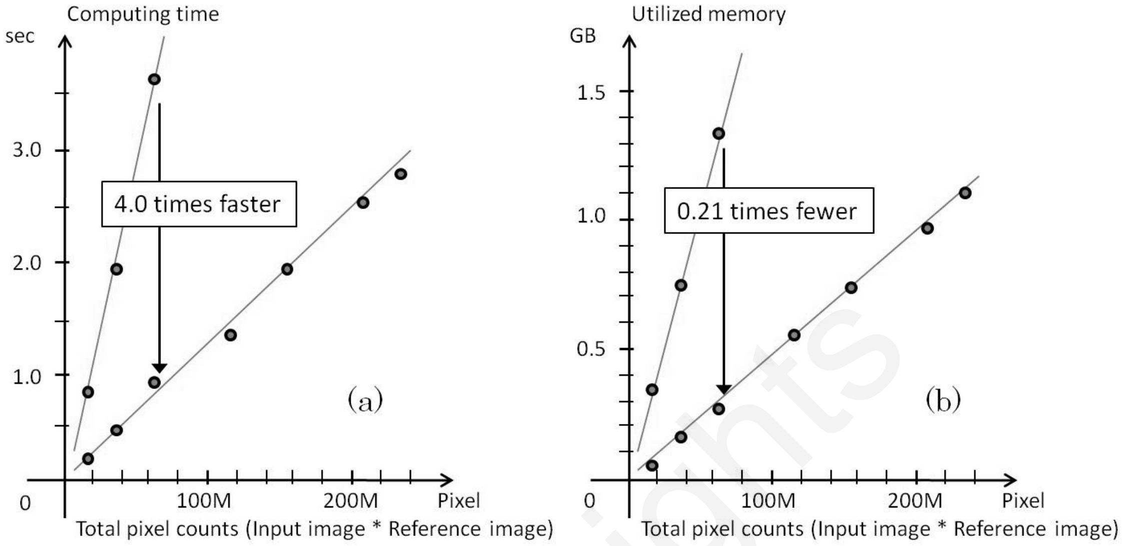


Figure 9. Results comparison of conventional and proposed 2DCDP Computing time comparison (a); Utilized memory comparison (b)

Copy Rights

6. CONCLUSION

In this paper, we report the result of the implementation of the optimization and acceleration methods for two-dimensional continuous dynamic programming (2DCDP) with memory reduction and parallel processing.

As a result of profiling, because the calculation costs on accumulation processing are particularly large, we have greatly improved the computation time for accumulation. The accumulation processing is parallelized using OpenMP in the same rank because each element in the same rank are mutually independent. Moreover, the search of the corresponding point from an input image to a reference image and accumulation processing are parallelized using OpenMP. As a result, processing is accelerated by dividing the calculation of $O(N^4)$ into $O(N^2)$ and executing it in parallel. In addition, the local cost buffer and the accumulation cost buffer are reduced by reviewing the processing. Finally, on the average, the calculation time is 4.0 times faster than the original time, and the memory cost becomes 21% of the original memory cost.

This method can be applied to acceleration on SIMD or cluster processing systems such as GPGPU and Cell processors by memory reduction and improvement of granularity.

REFERENCES

Dalal. N et al., 2005. Histograms of oriented gradients for human detection. IEEE Computer Society Conference on CVPR, Vol. 1, pp. 886-893.

Franchetti. F et al., 2009. Discrete Fourier Transform on Multicore. IEEE Signal Processing Mag, Vol. 26, No. 6, pp. 90-102.

Iseki. K et al., 2008. 3D Shape Reconstruction Using Optimal Pixel Matching Between Images. IPSJ SIG Notes. CVIM2008, pp. 101-108.

Iwasa. Y et al., 2005. Algorithm for guaranteeing monotonous contiguity of pixel correspondence in spotting recognition of image, MIRU 2005, pp. IS3-98.

Kawashima. Y et al., 2009. High Speed and High Accuracy Motion Vector Detection by On-sensor Pixel Matching. ITE Technical Report, Vol. 33, pp. 29-32.

Kim. H et al., 2009. Multicore Software Technologies: A Survey. IEEE Signal Proc Mag, Vol. 26, No. 6, pp. 80-89.

Lowe. D, 2004. Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision, Vol. 60, No. 2, pp. 91-110.

Machino. T et al., 2008. Optimizing Two-Dimensional Continuous Dynamic Programming for Cell Broadband Engine Processors, FCST 2008, pp. 186-193.

Nishimura. T et al., 1997. Two-dimensional Continuous DP for spotting recognition in an image. PRMU1997, pp. 1-7.

Ohtaet. Y et al., 1985. Stereo by Intra- and Inter-Scanline Search. IEEE Trans. on PAMI, Vol. 7, No. 2, pp. 139-154.

Oka. R, 1998. Spotting Method for Classification of Real World Data. The Computer Journal, Vol. 41, No. 8, pp. 559-565.

Okutomi. M et al., 1993. A Multiple-Baseline Stereo. IEEE Trans. on PAMI, Vol. 15, No. 4, pp. 353-363.

Pereira. S et al., 2000. Robust Template Matching for Affine Resistant Image Watermarks. IEEE Transactions on Image Processing,0 Vol. 9, No. 6, pp. 1123-1129.

Xiaojun. Z et al., 2002. Real-time 3D Shape Reconstruction and Refinement from Multi-viewpoint Image Sequences. IPSJ SIG Notes. CVIM2002, pp. 61-68.

Yaguchi. Y et al., 2008. Two-dimensional continuous dynamic programming for spotting recognition of image. Proc. Meeting on Image Recognition and Understanding (MIRU2008), pp. 707-714.

Zhu. S et al., 2000. A New Diamond Search Algorithm for Fast Block-Matching Motion Estimation. IEEE Transactions on Image Processing, Vol. 9, No. 2, pp. 287-290.

Copy Rights

EFFECTIVE RESOURCES ALLOCATION IN A P2P OVERLAY TO EXECUTE GRID WORKLOADS

Rocco Aversa, Luigi Buonanno, Beniamino Di Martino and Salvatore Venticinque
Department of Information Engineering - Via Roma 29, 81031 – Aversa, Italy

ABSTRACT

Peer to peer systems (P2P) allow users to be connected in order to share resources and information. They have been exploited in different application contexts: distributed computing, contents and files sharing, collaborative systems. Nevertheless many proposals have been conceived to exploit P2P for sharing computational power, however each known successful solution has been designed to solve a specific problem. Many issues arise if one aims at approaching the design of a general P2P platform to support parallel/distributed computing according a Grid-like philosophy. In this paper we propose a completely distributed approach. Upon an overlay that is similar to the most popular Kademlia based P2P file sharing systems, we aim at supporting decentralized sharing of computational resources for transparent and remote execution of user's applications. We focus on effective gathering of resources and on workload management of submitted tasks. Simulation results of different task allocation strategies are presented.

KEYWORDS

P2P, power, sharing, resources, balancing.

1. INTRODUCTION

A common motivation for P2P projects is the availability of billions of pcs over the network that, for the most part of the day, are performing nothing more than idle cycles. Even if the overlay configuration can change very quickly, the great number of peers can always grant the availability of needed resources. The main benefits provided by the utilization of a P2P architecture for resources sharing are: 1) *Cost reduction*, because only available pcs on the internet are used, rather than very expensive supercomputers; 2) *Performance improvement*, in fact there is a huge number of idle machines available over the Internet; 3) *Reliability*, because replicating a task, the more a system is distributed, the lesser it can be that a fault somewhere will compromise it.

Existing solutions do not work if we are looking for an infrastructure that allows users to exploit shared resources by delegating the execution of their applications to the network. That is because they are conceived to solve specific problems. For example Seti@home [20] adopts a centralized approach where a top layer keeps the scope of the problem, and a myriad of mindless nodes provide the raw power. It works at its best with parallelizable problems that are in charge of a single company or entity. For a different scenario, where ***n*** nodes are interested in having their tasks performed, and ***m*** nodes provide computing power, for example to get credits, a different solution is needed. Some questions to be answered are: how many peer should be used to schedule the task in order to bound overhead, to get desired performance and reliability? how choosing the best peers among the available ones? In this paper a completely decentralized approach for distributed scheduling is proposed and different task allocation strategies have been evaluated.

Mandatory facilities of a framework that implements such a P2P overlay are:

Effective resources lookup. Our overlay of nodes and resources retrieval is based on the well-known distributed *Hash Tables* algorithm. This allows a very efficient and predictable resource discovery and localization. The mechanism is very similar to the one adopted by the most popular, Kademlia based, P2P file sharing systems [12].

Profiling. In a traditional P2P system (file sharing-oriented), each peer shares a certain amount of files. Similarly, in our model every peer will share one or more computing profiles which describe software and hardware equipment of nodes. In this regard, we need to describe, in a synthetic but exhaustive way, the

Copy Rights

heterogeneous architectures that compose the overlay network. We investigate a model for resource characterization, that allows to describe peers and to discover them according to both architectural and performance parameters.

Performability. In a typical file sharing P2P environment, the same resource is usually shared by many peers. Redundancy allows to grant both availability (a node quitting won't compromise the download) and performance (a peer can download the resource by multiple sources, reducing the required time).

Other issues to be addressed are (but they are not limited to) resource localization, interoperability among heterogeneous systems, load balancing, resource optimization, security and trusting. We focus here on resources allocation strategies for an effective workload balancing and distributed scheduling. Effective collection of resources to optimize the system utilization and the performance of the distributed execution represents the main objective. The paper is organized as follows. In section 2 we discuss related works. Section 3 presents our P2P approach for computational resources sharing. Section 4 describes a model of our system and its characterization. Section 5 shows simulation results. Finally we come to conclusions.

2. RELATED WORKS

Peer to Peer (P2P) refers to logical organization of computing entities where each individual knows its neighbors and can behave both as a server and a client. We can distinguish three main classes of P2P applications: distributed computing oriented, file sharing and collaborative. In particular distributed applications split complex tasks into smaller sub-tasks that can be performed on a number of independent nodes spread over the net. There are some relevant examples of P2P systems oriented to parallel and/or distributed computing, which have been successful in their exploitation. Nowadays P2P represents an alternative approach to Grid [8] for resource sharing in heterogeneous and geographically distributed systems. GRID has been developed to support resource sharing among heterogeneous machines geographically distributed and administrated by different organizations. The objective of Grid is to exploit great part of them that is usually underutilized for most of the time when someone requires computing power for its scientific application [9]. P2P systems, on the other hand, have been developed in order to allow heterogeneous users to share information and with the purpose of optimizing the availability of these data in a dynamic environment where users can log in and out with a high frequency. [18] reports a comparison between the P2P and Grid approaches to distributed computing. A major difference is that the Grid computing is mostly used to aggregate very powerful, distributed and dedicated machines. On the other hand, the P2P approach relies on common, general purposes machines distributed across the Internet. Current P2P systems have the perk of allowing a very high number of users (hundreds of thousands is a common figure). Anyway, they offer few services, without doing assumptions on the reliability of the peers themselves [8]. Adaptivity is anyway the biggest benefit that P2P systems deliver. They can automatically adapt to changes in the environment, as connections, disconnections, network failures etc. On the other hand, without any sort of distributed scheduling, it is very complicated to ensure a given QoS level [7]. There are many projects which use a P2P paradigms in order to index available Grid resources. Distributed representation, indexing and search are addressed by the project DBGlobe [14]. As different peers have different hardware architectures, middlewares, different O.S., not every peer is a suitable candidate for a given task.

Performance in P2P system is another open issue. There are very few papers that adopt analytic models to analyze the performances of a P2P network [19]. In [16] a middleware has been showed capable of enabling the mutual and joint power sharing between users that hold heterogeneous computational units. Possible criteria adopted to group the peers are: distance, QoS and available resources [3]. In [15], a technique has been proposed to improve load balancing in a P2P system. In [11] an architecture has been described (CompuP2P) for the resource sharing on large scale networks. CompuP2P uses a protocol based on Chord [17] and detects a set of "dynamic markets", each of them groups all the peers that are willing to buy or sell the same "amount" of computing power. Anyway a special peer ("Market Owner), that is responsible for the association between requests and offers of computing power, represents a bottleneck. In [10] is proposed a solution for the scheduling of multiple applications in a concurrent fashion.. Authors propose a decentralized scheduling pattern and do a comparative analysis of different heuristic logics. There are many Grid solutions for task scheduling and workload distribution. For example Condor [6] is a high-throughput distributed batch computing system that provides a job management mechanism, scheduling policy, resource monitoring, and resource management. However, it can hardly be defined as a P2P system, cause of the presence of a central manager that accepts job submissions. The objective of our research is to design a P2P infrastructure in order to exploit not only Grid resources, but above all huge numbers of machines which connect dynamically to the network and do not provide any guaranties.

3. MODELING AND SIMULATION OF RESOURCE ALLOCATION

The model we propose is finalized to achieve a completely decentralized, high-throughput, distributed system. All the peers can behave like clients which submit jobs, and in the meanwhile they can download different kinds of jobs by other peers, disconnect and execute them, reconnect to retrieve asynchronously their results (and providing the output of their own computations). In our vision each user that joins a P2P overlay can delegate the execution of his applications to a pool of peers whose characteristics are compliant with the application requirements. On the other hand, he can share his own resources to get, eventually, credits toward the system. Every peer may perform both roles, also at the same time. Connected peers join a Kademlia [12] P2P network. Each server peer will be characterized by one or more profiles, which contain all the relevant information about his hardware and software configuration. Profiles represent what peers offer to the network overlay. Information include, but is not limited to CPU architecture, number and speed of processing units, memory size and speed, mass storage capabilities, software libraries, Operative System, middlewares (agent platforms, MPI, ...), costs. All the parameters are described according to a common ontology. Then, a digest of each profile is pushed into the P2P overlay. In this way, using the well-known, consolidated Kademlia algorithms, every client will be able to look for those profiles which are compliant with the requirements of its applications. Profiles are searched and downloaded as in any P2P file sharing systems.

Simulation allows the evaluation of the effectiveness of some resources allocation schemes using a P2P overlay for executing a Grid workload. We need to characterize the resources available in a real P2P system and the workload of a real Grid system. In Figure 1, our methodology is shown. We considered the logs of a P2P system for classifying profiles and behavior of machines which are available in the Internet. Performance information about execution of jobs in a Grid system have been analyzed to get a feasible statistic that describes arrivals and computational requirements of tasks which will feed the P2P network overlay. These statistic have been used to feed the simulation model.

Copyright

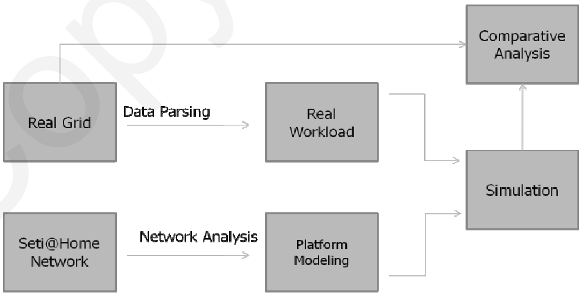


Figure 1. Simulation schema

We expect that simulation results can provide results about how task should be handled in a P2P computing system, and compared how performance are comparable with the ones provided by Grid.

3.1 The P2P Model

We suppose that infinite compliant servers are always available. Hence, it is necessary to select the best set of n peers which will be candidate to execute the task. We call it a pool. A task will be scheduled among the peers of the pool till its completion. Of course many schedule strategies can be implemented once the pool has been composed. We assume here that each server can join different pools. It maintains a private queue of tasks to be executed. Tasks are served in each queue according to a FCFS (first come, first served) policy. When a pool has been identified the task is replicated in the queues of all servers. In our model when a task is being to be executed the server notifies this event to the others, which delete the task from their queues. In this way, we avoid that two or more servers execute the same task at the same time. Here, just for simulation purposes, we assume that neither faults, nor disconnections happen meanwhile the task is executing. Different choices could be considered to grant reliability or to increase the probability of completion. In order to optimize the system performance, it is very important to detect the optimal dimension of the pool (the n value) and the policy for the distributed scheduling. *(1)* represents the function that finds **n** available resources, which are suited to execute the task **t**.

$$c: t \in T \rightarrow P_i = [\![r_1, \dots, r_n]\!] \in R^n \qquad (1)$$

$r_1$is a resource that is available in the P2P network and satisfies the requirements for the execution of a task t. Resources, which have been selected using the *c* function, will be chosen among the ones that, at least and not exactly, satisfy the minimal requirements for executing t. We expect that less demanding tasks will benefit of big values of n. In fact they will be replicated also in powerful peers, which will have their queues increasingly crowded because of many less demanding tasks together with few more demanding tasks. On the other hand, n values beyond a threshold do not improve performances and overload the peers and the network. The P2P overlay has been modeled as a set of M/M/k/∞/∞ queues. K is the number of available machines which have similar features, and above all provide comparable computing power. A pool of server hosts is modeled as a set that is composed of one or more multiple server queue (it means same service time μ in the model). Task sources are modeled as Poisson distributions with different arrival rate (λ) and different computational requirements (σ). The goodness-of-fit evaluation results, obtained with the Kolmogorov-Smirnov test, in comparison of the model (job interarrival times is a poisson distribution) to the real workload are very promising. Furthermore, we have modeled the input workload as sum of poisson distributions because a sum of two poisson distributions P(λ1) and P(λ2) is equivalent to P(λ1 + λ2). To feed the model we evaluate in the following the λ and σ parameters for the tasks which are executed in a real Grid, and μ for a real P2P overlay. Faults of nodes and disconnections have not been considered (in simulations), but reliability can be improved by increasing the number of peers which are scheduling the same task.

3.2 Characterization of P2P Computing Resources for Simulation

To achieve a significant analysis of the proposed platform, we referred to the current hardware infrastructure of the Seti@home project. The Boinc team keeps an updated statistic of the projects that includes members and individual contributions, and makes it publicly available. They monitor different kind of data (CPU type, number of hosts, total credits, total average credits). Credits (or Cobblestones) are the units used by the various Boinc projects to track the amount of computational work performed by a peer executing a given task. Its name is due to Jeff Stone of the Seti@home team. The main concept is that 100 cobblestones are awarded for a day of work on a computer capable of:

- 1000 double-precision MIPS based on the Whetstone benchmark;
- 1000 VAX MIPS based on the Dhrystone benchmark.

In other words a pc can provide 10000 cobblestones, or credits, if it can perform 105 double precision MIPS in a day, and 105 VAX MIPS. This parameter takes implicitly into account both the raw computing power of a computer, and the fraction of time it is available for executing tasks from the P2P overlay. In the simulated environment, we will use the Boinc credit system, and more specifically the RAC (Recent Average Credits) as a synthetic and mono-dimensional parameter to characterize peers capability. Notice that this is only for simulation purposes: in the framework we are developing, a profiling mechanism has been implemented to match the requirements of tasks with available resources.

Recent Average Credits is an estimation of how many credits a computer of a given class can, in average, earn in a day. We have selected the main 200 typologies of computers that contributed to the overall recent credits sum. This allows us to design a realistic scenario. The selected computer categories have been grouped according to 9 macro classes, and for every class an average computing power has been calculated.

Table 1 shows for each class the number of nodes belonging to it, number of credits provided by the full class, the number of credits provided daily by each cpu and the minimum and maximum threshold used to assign an host to that class.

Table 1. Classification of peers according to the credits they provide to the P2P overlay

CAT.	Hosts	Recent Avg. Credits	Recent Avg. Credits per Cpu	Min Rac	Max Rac
A	291	410298	1409,958763	1000	2000
B	8427	6224000	738,578379	500	1000
C	34595	12683643	366,632259	250	500
D	111134	19722944	177,4699372	125	250
E	127379	10158572	79,75075954	60	125
F	180630	7655528	42,38237281	30	60
G	175426	4065000	23,17216376	15	30
H	413686	4529068	10,9480814	7,5	15
I	351955	1770492	5,030449915	0	7,5

3.3 Synthetic Representation of a Grid Workload

The next step has been the evaluation of workloads submitted to real Grid networks. For this purpose, we exploited the data collected by the project Grid Workload Archive (http://gwa.ewi.tudelft.nl). The project makes available traces belonging to many different grid systems, including arrival and execution time, used resources and cpu time. Across the grid systems that are part of the project, we focused our attention on Auvergrid (http://www.auvergrid.fr) because it is characterized by simple workload (all the jobs are sequential) and homogeneous CPUs (all the hosts are equipped with Xeon DP 3,0 Ghz).

Table 2. Synthetic representation of Grid workload

Percentage	Duration	Average (credits)	Jobs per minute
0,25	< 1min	0,071141	2
0,15	<10min	0,782551	1,2
0,15	<100min	7,82551	1,2
0,2	<1000min	78,2551	1,6
0,25	<10000	177,8525	2

Copy Rights

4. SIMULATION RESULTS

The tool used is Java Modeling Tools[1,2]. It allows to model and simulate queue network and to collect and represent resulting statistics. Figure 2 shows a schematic representation of the implemented queue network.

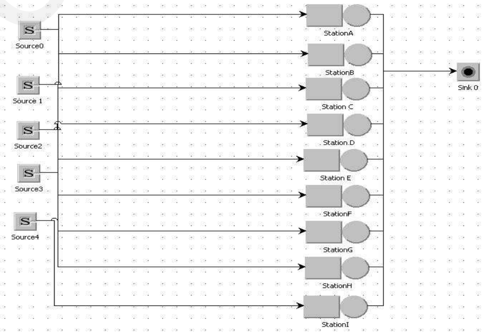


Figure 2. Queue network implemented in JMT

Sources are the points where tasks are injected into the system. Stations are composed by a variable number of servers. The simulation stops when a confidence interval of 0.95 has been obtained, with a max relative error of 0.05. Tests were performed under different conditions and assumptions, as discussed in the next subsections. Notice that, being the Auvergrid network rather small (450 hosts) if compared with the resources available in the Boinc P2P overlay, we decided to feed the queue network with an increased number of job sources, as it expected in a real system. It is equivalent, dealing with Poisson distributions, to multiply the arrival rate for the same factor. Tests has been performed with different multiplicative factors x. The results presented in this paper have been obtained with x=10. A server in the model is characterized by a set of service times μ_i, one for each task class.

4.1 Allocation Strategies

We have to consider that usually P2P and Grids provide a best effort service. It means that it is not possible to foresee when a job will be scheduled and how much time it will be used for its completion. Any strategies chosen to allocate resources to tasks could use only heuristics. The c function described by (1) is implemented Figure 3 (a,b,c).

	Source 0	Source 1	Source 2	Source 3	Source 4
A	√				
B	√	√			
C		√			
D			√		
E			√		
F				√	
G				√	
H				√	
I					√

(a)

	Source 0	Source 1	Source 2	Source 3	Source 4
A	√	√			
B	√	√			
C		√	√		
D			√		
E			√	√	
F				√	
G				√	
H				√	√
I					√

(b)

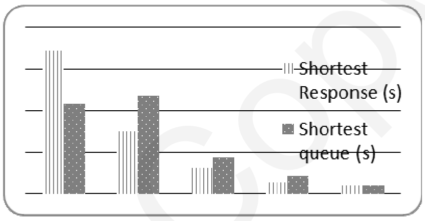
	Source 0	Source 1	Source 2	Source 3	Source 4
A	√	√	√	√	√
B	√	√	√	√	√
C	√	√	√	√	√
D	√	√	√	√	√
E	√	√	√	√	√
F	√	√	√	√	√
G	√	√	√	√	√
H	√	√	√	√	√
I	√	√	√	√	√

(c)

Figure 3. Allocation strategies

4.2 Small Fit Allocation Strategy

In this scenario, the tasks incoming by different sources are allowed to be put in queue according to the rules listed in Figure 3.(a). Light jobs are sent only to less powerful nodes. In the same way, heavy computational tasks will only be queued on machines that can, theoretically, ensure an execution within a certain time. As a general rule, we performed the simulations using two different policies (Join Shortest Queue and Join Shortest Response). Results are shown in Figure 4.



(a) Bar chart

	Shortest Response (s)	Shortest queue (s)
task0	347	218
task1	151	237
task2	63	87
task3	27	44
task4	20	20
System average	129,25	116,9

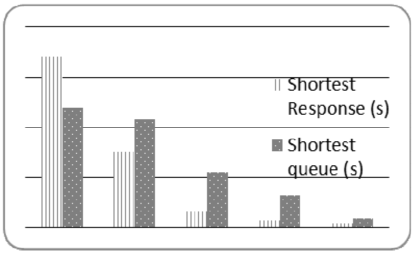
(b) Time values

Figure 4. Task turnaround for best fit allocation

4.3 Medium Fit Allocation Strategy

Task per peer category association rule defined in Figure 3(b) make available a greater number of nodes for the execution of light tasks. The simulation test shows a degradation of the performance for tasks belonging to classes of jobs that had to share their peers with new less power-demanding categories. Notice that the turnaround of tasks belonging to class 2, in this scenario, are deeply influenced by the routing algorithm. As it is shown in Figure 5, using a Shortest Response routing the turnaround is 50% of the previous one. Using a Shortest Queue routing, instead, the turnaround is doubled.

Copy Rights



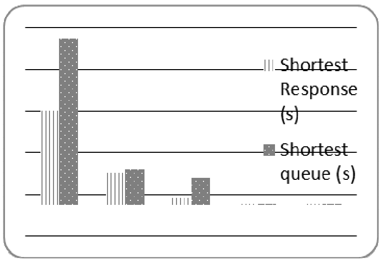
	Shortest Response (s)	Shortest queue (s)
task0	340	238
task1	151	215
task2	31	111
task3	14	64
task4	9	18
System average	117,35	125,7

(a) Bar chart (b) Time values

Figure 5. Task turnaround for Medium Fit Allocation

4.4 Large Fit Allocation Strategy

The last simulated scenario, provides the behavior of the system when every peer is allowed to join a queue of any peer belonging to any class (Figure 3(c)). In this case, it is straightforward that performances of most power demanding jobs get dramatically worse, while the lightweight jobs benefit of a big improvement in their performance. The values for the intermediate classes remain basically unchanged. Performance figures are shown in Figure 6.



	Shortest Response (s)	Shortest queue (s)
task0	453	800
task1	153	170
task2	31	130
task3	1,5	4
task4	0,13	0,3
System average	141,1825	245,875

(a) Bar chart (b) Time values

Figure 6. Task turnaround for Large Fit Allocation

Copy Rights

5. CONCLUSIONS

A preliminary analysis of experimental results let us draw some general considerations on the effectiveness of P2P system for executing Grid workload. Firstly, a Join the shortest queue approach, despite being easier to implement, is more influenced by the design of the *n* parameter. Comparing the first and third examples, it is clear that the system average response time has doubled, and it is even increased by a factor 4 for the tasks belonging to the task0. Anyway, a proper setting of the task-peer association table can lead to comparable performances for the two routing algorithms. Another clear result is that highest and lowest demanding classes are the ones most sensible to the dimensioning of the number of servers n, while middle classes receive a lower impact by it. By a qualitative point of view is quite predictable if we consider the task0 and task4 : increasing n, jobs of task0 will have to share the suitable peers with jobs of the other classes, while not having more peers added to their list. On the other hand, tasks of task4 will be able to join the queues of more peers, instead of be limited to overcrowded and slower hosts. Furthermore, only for simulations, we assumed that there were not failures due to fault or to unexpected disconnections.

Future works will lighten some limitations of the model: communication among peers, the time needed to discover on-line servers, a finer control of the number of server that will be less than all the ones belonging to the same class. We are investigating dynamic scheduling for this kind of environments because there's no chance to foretell in advance how much time a node will be connected, collaboration among peers for the execution of parallel job. We are extending a distributed algorithm, previously conceived in [4,13], for task scheduling based on game theory. We also plan to evaluate some alternative approach where calls for task execution are shared and computing nodes compete to answer.

ACKNOWLEDGE

This work has been supported by PRIST 2009, “Fruizione assistita e context aware di siti archelogici complessi mediante terminali mobile”, founded by Second University of Naples.

REFERENCES

1. Bertoli, M.et al, 2009.Jmt: performance engineering tools for system modeling. SIGMETRICS Perform. Eval. Rev. 36, pp 10-15

2. Bertoli, M. et al, 2006. Java modelling tools: an open source suite for queueing network modelling and workload analysis. In: Proceedings of QEST 2006 Conference, Riverside, US, IEEE Press, pp 119-120

3. Bourgeois, J. et al, 2004. Using similarity groups to increase performance of p2p computing. In 10th Int. Euro-Par Conference (Europar04), volume 3149 of LNCS, Springer (2004), pp 1056-1059

4. Chapman, A.C. et al, 2009. Decentralised dynamic task allocation: a practical game: theoretic approach. In: AAMAS '09: Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems, Richland, SC, International Foundation for Autonomous Agents and Multiagent Systems, pp 915-922

5. Douglas Thain and Miron Livny,2003. Building Reliable Clients and Servers. In Ian Foster and Carl Kesselman, editors, The Grid: Blueprint for a New Computing Infrastructure, Morgan Kaufmann, 2003, 2nd edition. ISBN: 1-55860-933-4.

6. Douglas, T. et al, 2005. Distributed Computing in Practice: The Condor Experience. Concurrency and Computation: Practice and Experience, Vol. 17, No. 2-4, pages 323-356

7. Drost, N. et al, 2006. Simple locality-aware co-allocation in peer-to-peer supercomputing. In: CCGRID '06: Proceedings of the Sixth IEEE International Symposium on Cluster Computing and the Grid, Washington, DC, USA, IEEE Computer Society,pp 14

8. Foster, I., Iamnitchi, A.,2003. On death, taxes, and the convergence of peer-to-peer and grid computing. In 2nd International Workshop on Peer-to-Peer Systems (IPTPS03), pp 118-128

9. Garcia, F.D., henk Hoepman, J.,2004. Offline karma: Towards a decentralized currency for peer-to-peer and grid applications . In Workshop on Secure Multiparty Computations

10. Ghatpande, A. et al, 2008. Analysis of divisible load scheduling with result collection on heterogeneous systems. IEICE Transactions 91-B, pp 2234-2243

11. Gupta, R., Sekhri, V., Somani, A.K., 2006. Compup2p: An architecture for internet computing using peer-to-peer networks. *IEEE Trans. Parallel Distrib. Syst. 17*, pp 13061320

12. Maymounkov, P., Mazières, D., 2002. Kademlia: A peer-to-peer information system based on the xor metric. In: IPTPS '01: Revised Papers from the First International Workshop on Peer-to-Peer Systems, London, UK, Springer-Verlag, pp 53-65

13. Micillo, R.A. et al, 2009. A grid service for resource-to-agent allocation. In: Internet and Web Applications and Services, International Conference on. Volume 0., Los Alamitos, CA, USA, IEEE Computer Society, pp 443-448

14. Pitoura, E. et al, 2003. Dbglobe: a service-oriented p2p system for global computing. SIGMOD Rec. 32, pp 77-82

15. Saruladha, K., Santhi, G., 2007. Behavior of agent based dynamic load balancing algorithm for heterogeneous p2p systems. Computational Intelligence and Multimedia Applications, International Conference on 1, pp 109-113

16. Shudo, K. et al, 2005. P3: P2p-based middleware enabling transfer and aggregation of computational resources. Cluster Computing and the Grid, IEEE International Symposium on, pp 259-266

17. Stoica, et al,2001.Chord: a scalable peer-to-peer lookup protocol for internet applications. *IEEE/ACM Trans. Netw.* 11, pp 17-32

18. Tang, J., Zhang, M.,2006. An agent-based peer-to-peer grid computing architecture: convergence of grid and peer-to-peer computing. In: ACSW Frontiers '06: Proceedings of the 2006 Australasian workshops on Grid computing and e-research, Darlinghurst, Australia, Australian Computer Society, Inc., pp 33-39

19. Yang, X., de Veciana, G,2006.: Performance of peer-to-peer networks: service capacity and role of resource sharing policies. Perform. Eval. 63, pp 175-194

20. Werthimer,D. et al, 2001. Seti@home: massively distributed computing for Seti. Computing in Science and Engg.Vol 3, pp 78-83

Copy Rights

DESIGN AND DEVELOPMENT OF CONSTRUCTIVIST EDUCATIONAL SOFTWARE TO DEAL WITH STUDENTS' EMPIRICAL IDEAS ABOUT BASIC OPTICS CONCEPTS

Tekos George and Solomonidou Christina
University of Thessaly

Copy Right

ABSTRACT

The paper introduces the design and development of quality interactive multimedia educational software based on students' empirical ideas and conceptual difficulties, identified in Greek students 7–12 years old. The software promotes interdisciplinary study of geometrical optics concepts. We present a survey, which investigated 40 students' initial ideas about light phenomena using interviews. Then we describe the design of the educational software 'Light-Life', which was designed based on constructivist views of learning. The concepts analysed in this study were linear propagation of light, shadows formation, light reflection, diffusion and refraction, synthesis of colour light beams, and vision. Appropriate printed worksheets were also developed for the students. The proposed approach intends to improve the quality of educational approach and tools to better respond to students' learning with understanding and help them reformulate their empirical ideas to better explain everyday life situations related to basic optics phenomena. In this direction, the preliminary evaluation research had positive results regarding students' learning with understanding.

KEYWORDS

Multimedia educational software, constructivism, primary students' conceptions, conceptual change, optics phenomena, interdisciplinary teaching

1. INTRODUCTION

Science education research has revealed that the majority of students enter school with pre-instructional knowledge or beliefs about natural phenomena and concepts based on their everyday experience. Their personal views about science phenomena integrate into students' cognitive structures and contradict science concepts universally accepted by the scientific community. They develop only a limited understanding of science concepts following instruction (Driver and Oldham, 1986; Driver et al., 2000). Furthermore, it is possible that students may apply scientific ideas in solving traditional science text-book problems in school examinations, but not in explaining natural phenomena in everyday life (Driver, 1989; Driver et al., 2000). So it is essential for teachers to become aware of their students' conceptions and misunderstandings in order to organize their teaching more effectively.

The emergence of constructivist views of learning promises to improve learning and teaching in school. Constructivism is viewed as a theoretical perspective about knowledge construction, which may be useful to the design of constructivist learning environments (Jonassen, 1999).

Educational software has great potential as a cognitive tool (Jonassen, 1993). Although, this offers a powerful environment for studying formal representations, its actual contribution depends on how effectively each task is designed in order to enhance student achievement (Bransford et al., 2000; Tekos and Solomonidou, 2009). Conceptual difficulties are a prerequisite for designing and developing effective instructional approaches utilizing the potential of the information and communication technology (ICT) tools. It is necessary to investigate and take into account students' empirical ideas before designing educational activities and selecting appropriate ICT tools in teaching (Solomon, 1994; Osborne, 1996; Jonassen, 1999).

This study is based on the D.E.S.T.E. model (Solomonidou, 2006), which describes the steps that should be followed to create, implement and evaluate constructivist learning environments with the use of ICT tools. The name comes from the initials of the Greek words for Investigation, Conception, Design, Development, and Implementation, as follows: a) Investigation of students' initial empirical conceptions, b) Conception of the teaching and learning content based on both the scientific knowledge and the students' initial empirical conceptions and conceptual needs, c) Design of constructivist learning environments which are student-centred, collaborative, problem solving and authentic task-based, and supported by teacher scaffolding, d) Development and formative evaluation of the educational environment, e) Implementation of the digital environment in the classroom and final evaluation of it based, among other things, on students' final conceptions and learning outcomes.

2. INVESTIGATION OF STUDENTS' IDEAS

2.1 Previous Studies

For almost four decades there has been intensive research directed to students' alternative conceptions regarding light phenomena across all ages (Andersson and Karrqvist, 1983; Guesne, 1985; Galili and Hazan, 2000; Osborne et al., 1993; Selley, 1996; Watts, 1985).

Concerning the nature of light, students do not conceive light as a distinct entity. Some of them equate light with a source and others with its effect. So they have difficulty in interpreting a range of light-related phenomena (Guesne, 1985). For seeing in the darkness, students do not recognize the necessity of light and think that it is possible to see things even if it is dark. They do not consider the presence of light as the essential factor in order for them to see things even faintly, explaining that eyes can get used to seeing in total darkness (Fetherstonhaugh and Treagust, 1992). They claim we can see things just 'because our eyes have the ability to see' or 'because objects are bright' (Tiberghien et al., 1980; Andersson and Karrqvist, 1983; Osborne et al., 1993; Ravanis et al., 2002).

Concerning light propagation students think that the distance travelled by light varied from a few millimetres to an infinite distance. Most children decided that the distance travelled by light depended upon whether it was day or night (Fetherstonhaugh and Treagust, 1992), and only one direction from each source, like flash light beams (Bendall et al., 1993).

Moreover, the majority of 10 - 12 year old students think that shadow belongs only to a non-luminous object and that always looks like the object (Feher and Rice, 1988). Students tend to believe that shadow is not absence of light, but the presence of something tangible to which they give material characteristics (Bendall et al., 1993). According to Galili and Hazan (2000) children perceive shadows in much the same way as optical images. Shadows can be manipulated in the same way as independent objects.

Most of the students think that in the region of geometrical overlap there would be either lightness (full illumination) or darkness (shadow). They do not consider semi-darkness. Also, many children aged 11-12 believe that light stays inside a mirror or on a piece of paper when it falls on it (Guesne, 1985).

Regarding vision primary students do not believe that their eyes receive light when they look at an object. A great number of students, generally younger ones, attribute no relationship between object, light and eye (Osborne et al., 1993; Ravanis, 1999), in spite of the well known fact that we can see objects because of the presence of ambient light. Other students think that we can see due to a 'light bath' that fills space and they draw simple connecting lines without showing direction between the vertexes of the classical triangle: source-object-eye (Hosson and Kaminski, 2002). Some students believe that we can see an object because the observer directs sight lines toward the object, with light possibly emitted from the eyes (Langley et al., 1997; Tekos, et al., 2008). Moreover, a difference between seeing luminous and non luminous objects has been indicated (Guesne, 1985). Students might adopt an 'active role' of the eye emitting light and receiving light in the case of luminous objects. Regarding colour, the majority of children think it is a property of objects, e.g. a book is red because has the ability to be red and has no relation to light (Fetherstonhaugh et al., 1987).

2.2 Our Study

During the year 2008–2009 we conducted the initial research with 40 Greek primary school students aged 7–12 years aiming at (a) investigating their initial ideas about light propagation, shadows formation, light reflection, diffusion and refraction, synthesis of colour light beams, and vision, and (b) informing the design of appropriate educational software and other didactic material to promote a better conceptual grasp of the subject. First, an open-ended questionnaire, with free response, was administered to 140 primary school students (6-12 years old) in order to identify students' empirical conceptions, which could serve as guide for interviews. After studying the initial findings we developed four semi-structured interview protocols, each one for students of 2nd grade (aged 7), 4th (aged 9), 5th (aged 11), and 6th grade (aged 12), based on students' alternative conceptions found in the initial search. Then we conducted individual clinical-type (Piaget, 1928) interviews with 40 students (ten of each grade). The analysis of the students' answers allowed us to identify the following fundamental alternative conceptions, some of which were revealed in previous studies (Andersson and Karrqvist, 1983; Osborne et al., 1993; Ravanis et al., 2002):

1st category: A great number of students, generally from 2nd grade think that light equals both to its source and effect, and that light is not conceived as a spatial entity propagating through space.

2nd category: Students, even in 6th grade, think that light reflection and light diffusion are phenomena, which happen independently of the kind of the surface light falls on. They do not know about the trajectory of a light beam falling on a smooth surface, such as a plane mirror. Moreover, a number of students pointed out that the light beam returns to the light source independently of the angle of incidence, and others stated that light stays on a plane surface. Light diffusion does not happen in the atmosphere for many students, who used to give special attributes to light rays, such as their inability to travel in space. Students also think that earth daylight is due to the existence of the sea, ozone, etc., and not to light scattering on particles, dust, etc.

3rd category: Primary students, up to 4th grade, conceive shadows in the same way as independent objects. Light was associated with shadow formation, mainly in the sense that a light source was mentioned verbally. Also, regarding the size of a shadow, they associated it with the brightness of the light source: the brighter the light source is, the bigger the shadow formation becomes. Moreover, when students of the same age were asked to draw the object's shadow with the light falling on it slantwise, many of them did not put in the same line source, object and shadow.

4th category: Students in 6th grade attribute the denaturation of the objects' shape (e.g. when a pencil is half sunk in water) to the shape of the object or they give material characteristics to objects, rather than different speed of light propagation into different material. Other students used refraction to mean reflection.

5th category: Regarding vision, the 'emission model' (the eye emits rays) is the dominating model among 4th and 5th grade students explaining how we can see non-luminous objects. Moreover, there seems to be no awareness of the directionality of light in sight processes in younger students, who could not represent light at all but rather illustrating the geometrical connection between the viewed object and the eye.

In order to cope with these students' learning difficulties we developed the software 'Light-Life' on the basis of these research outcomes. The software comprises visualisations, simulations, and learning activities, having an interdisciplinary character. As a matter of fact, Optics is essentially an interdisciplinary subject. Physics, biology, physiology, chemistry and psychology are all needed for comprehensive discussions of optical phenomena (Feynman et al., 1964; Gregory, 1979; Ronchi, 1970). Moreover, as Galili and Hasan *stated, 'optics instruction using only physics is limited and cannot confront spontaneous knowledge about light. Such instruction cannot explain those natural phenomena which intrigue the novice learner'* (Galili and Hasan, 2000, pp. 60). The instruction used in this study utilized a scaffolding process to guide the learner from what is presently known to what is to be known. Therefore, the student engages in cognitive processes, appropriate for the learner's zone of proximal development (Vygotsky, 1978).

2.3 The Design of the Digital Learning Material

We have designed and developed the educational software 'Light-Life' using Microsoft Visual Basic 6 as an authoring tool. The aim is to involve students in a technologically rich multimedia learning environment posing educational tasks and to provide help and feedback while undertaking a variety of investigations. They can come across concepts related to sound, energy, heat and related concepts, such as space, time and change. In the activities there is also an attempt to see light in other frameworks such as those of biology, medicine, linguistics, history, ethnography, and art. At the same time, students are encouraged to work together on a range of situations and problems concerning light phenomena.

'Light-Life' is multimedia software in the sense that it presents the user with various combinations of texts, static and moving pictures, sound, video, simulations, applets, etc. The way that the learning experience unfolds depends on the choices made by the user as he or she navigates his/her way through the multimedia environment. The structure of that environment is in some parts linear and in others tree-like. Along the linear sections the user progresses along a predetermined series of stages. In the sections with a tree-like structure the user has access to related sections to find the information he or she seeks. There is a main menu with seven sections, each of which takes the user to an introductory page providing access to many other pages by activating hyperlinks of his/her choice.

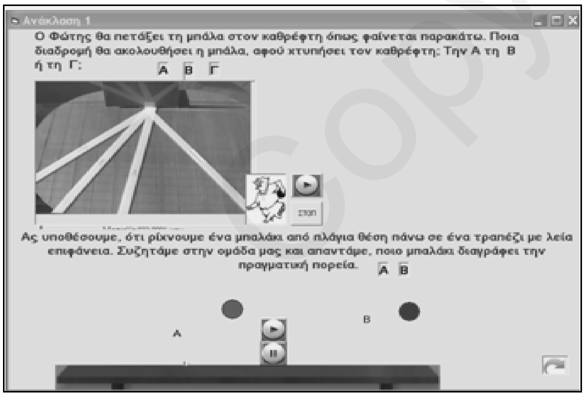
More particularly, 'Light-Life' consists of 95 screen shots aiming to help students construct knowledge according to the scientific accepted one, through various experiments. The software has a uniform design throughout the forms and simple and convenient navigation panels providing an easy manipulation of it. Specific questions are posed to students, having the following order: a) Eliciting their own interpretations or hypotheses about the phenomena they observe, b) use of metaphors relating to common everyday situations in order to help students construct suitable analogies, c) engaging students in activities to confirm -or not- their initial hypothesis, and d) extract deductions about the phenomena. The software provides immediate feedback to confront students' alternative ideas and help them redefine their hypotheses about the light phenomena. Moreover, it is accompanied with supplementary 'instructional drill and practice' activities. Useful tool tips pop up which help students to choose the right answer. When the student chooses a wrong answer a warning appears, pointing out that s/he has not taken into account something that had already been elaborated in a previous section.

Regarding the software's structure, in the first screen shot students can choose their level to go directly to one of the following five sections: experiments, video, glossary, and important people who studied Optics. The four levels correspond to four different school grades, 2nd, 4th 5th and 6th. The material presented to the user draws interdisciplinary concepts from the academic disciplines of Physics, Biology, Astronomy, Technology, Art, Medicine, and Literature.

2.4 Specific Features of the Software Items

The software 'Light-Life' aims to cope with students' alternative conceptions, found in the initial research and grouped in five categories as mentioned before, by using a series of features which are the following (for each category of students' alternative conceptions a specific feature is described):

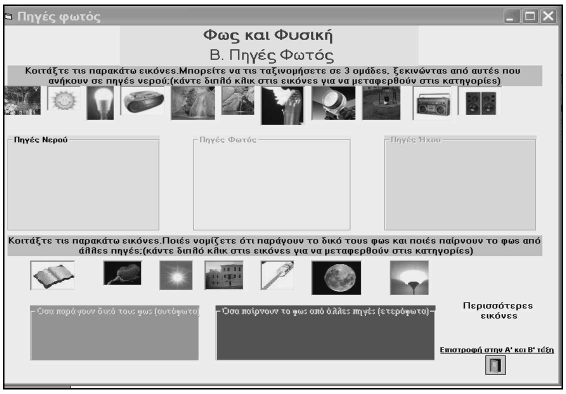
First category: Equating light with its source and effects was identified as the most prevalent students' initial idea among 7-8 years old students. This property of light is taken for granted in Greek school textbooks and yet is a prerequisite for understanding light in a more advanced level (Watts, 1985), as students do not distinguish between light as a physical entity and a sense perception stimulus.



Ο Φώτης θα πετάξει τη μπάλα στον καθρέφτη όπως φαίνεται παρακάτω. Ποια διαδρομή θα ακολουθήσει η μπάλα, αφού χτυπήσει τον καθρέφτη; Την Α τη Β ή τη Γ;

Ας υποθέσουμε, ότι ρίχνουμε ένα μπαλάκι από πλάγια θέση πάνω σε ένα τραπέζι με λεία επιφάνεια. Συζητάμε στην ομάδα μας και απαντάμε, ποιο μπαλάκι διαγράφει την πραγματική πορεία.

Figure 1. Real life video and animations of a ball bouncing on a Plane Surface



Φως και Φυσική

Β. Πηγές Φωτός

Κοιτάξτε τις παρακάτω εικόνες.Μπορείτε να τις ταξινομήσετε σε 3 ομάδες, ξεκινώντας από αυτές που ανήκουν σε πηγές νερού;(κάντε διπλό κλικ στις εικόνες για να μεταφερθούν στις κατηγορίες)

Κοιτάξτε τις παρακάτω εικόνες.Ποιές νομίζετε ότι παράγουν το δικό τους φως και ποιές παίρνουν το φως από άλλες πηγές;(κάντε διπλό κλικ στις εικόνες για να μεταφερθούν στις κατηγορίες)

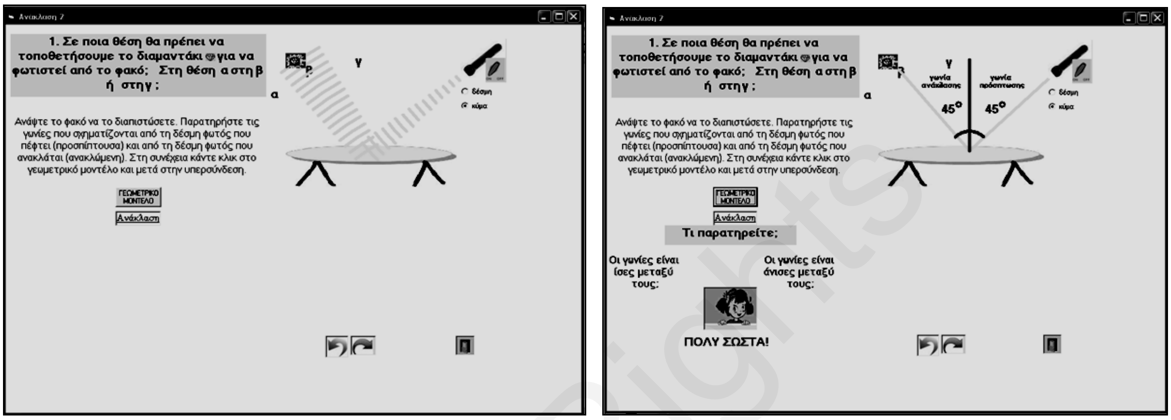
Όσα παράγουν δικό τους φως (αυτόφωτα)
Όσα παίρνουν το φως από άλλες πηγές (ετερόφωτα)
Περισσότερες εικόνες
Επιστροφή στην Α' και Β' τάξη

Figure 2. Drag-and-drop activities referring to various sources of light, sound, etc.

The software 'Light-Life' includes the following features to cope with these students' difficulties: A real life video about a ball bouncing on a plane surface and animations with tennis balls bouncing on a table used to help students construct a suitable analogy with light reflection. Students are asked to predict the 'correct' reflected course and then to confirm their prediction by activating the animation or the video (Figure 1). Gradually, a torch substitutes the child who throws the ball and a light beam substitutes the ball. Also, a number of drag-and-drop activities refer to various sources (sound, heat, energy or light sources) aiming to help students understand that a radio speaker differs from the sound it emits, an electric burner differs from heat, and light differs from a torch or the sun or any other light source (Figure 2).

Moreover, in another screen shot aiming to study the linear propagation of light, we use the following analogy: the student is asked to choose the shortest route a child must take to reach a given destination, i.e. a straight line. Then the child is replaced by a torch and a light beam falls on the spot that was previously the child's destination.

Second category: The initial students' answers showed that they do not distinguish between the kinds of surface on which they can observe light reflection and diffusion phenomena (i.e. plane mirror, cloth, shiny marble, ground, dust, etc.). Also they do not know about the trajectory of a light beam falling on a smooth surface such as a plane mirror. Moreover, a number of students pointed out that the light beam returns to the light source independently of the angle of incidence, and other ones stated that light stays on a plane surface. A number of activities with 'Light-Life' may engage students to observe the phenomena of light reflection and diffusion on different kinds of surface (Figure 3). The amount of light and the way it reflects on an object largely depends upon the smoothness or texture of its surface.



1. Σε ποια θέση θα πρέπει να τοποθετήσουμε το διαμαντάκι για να φωτιστεί από το φακό; Στη θέση α στη β ή στη γ;

Ανάψτε το φακό να το διαπιστώσετε. Παρατηρήστε τις γωνίες που σχηματίζονται από τη δέσμη φωτός που πέφτει (προσπίπτουσα) και από τη δέσμη φωτός που ανακλάται (ανακλώμενη). Στη συνέχεια κάντε κλικ στο γεωμετρικό μοντέλο και μετά στην υπερσύνδεση.

1. Σε ποια θέση θα πρέπει να τοποθετήσουμε το διαμαντάκι για να φωτιστεί από το φακό; Στη θέση α στη β ή στη γ;

Ανάψτε το φακό να το διαπιστώσετε. Παρατηρήστε τις γωνίες που σχηματίζονται από τη δέσμη φωτός που πέφτει (προσπίπτουσα) και από τη δέσμη φωτός που ανακλάται (ανακλώμενη). Στη συνέχεια κάντε κλικ στο γεωμετρικό μοντέλο και μετά στην υπερσύνδεση.

Τι παρατηρείτε;

Οι γωνίες είναι ίσες μεταξύ τους;

Οι γωνίες είναι άνισες μεταξύ τους;

ΠΟΛΥ ΣΩΣΤΑ!

Figure 3. The path of a light beam interacting with an object is demonstrated by turning on the light source simulating the relevant phenomenon considering the angle of incidence

Copy Rights

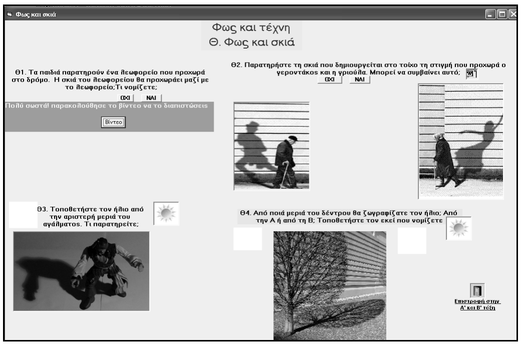
When the surface imperfections are smaller light reflects according to the Law of Reflection. Also, students can activate the applet http://micro.magnet.fsu.edu/primer/java/reflection/specular/index.html, which initializes with a beam of white light being reflected on a plane or rough surface demonstrating diffuse reflection. They can use slider bars to adjust the texture of the surface appearing in the window between a range of 0 percent (smooth) and 100 percent (maximum roughness).

Third category: In order to cope with students' misunderstandings about shadows the software 'Light-Life' includes the following features: For the 2nd grade there are the following sub-sections.

Sun and night and day alteration: Students go on a virtual journey into space to a point from which they can observe the earth in relation to the sun. They are asked to make hypotheses about the reason they see the earth half in darkness and half in light. They can also change the position of the earth in relation to the sun to observe the change in brightness of an area of the earth.

Light and art: This subsection includes shadow-creating games and other activities aiming to identify the objects that correspond to a series of shadows and also to give the correct orientation of the object's shadow in relation to the object and the light source (Figure 4, 5).

For the 4th grade, students can carry out some virtual experiments on the orientation of shadows (Figure 6). The students are asked to predict where the shadow of an object will be formed, in one case with one light source and in another with two light sources. They can then test their predictions by switching on the virtual light sources, overturning any misconceptions they may previously have had. The initial investigations showed that students assumed that the size of a shadow would be relative to the brightness of the light source.



Θ1. Τα παιδιά παρατηρούν ένα λεωφορείο που προχωρά στο δρόμο. Η σκιά του λεωφορείου θα προχωράει μαζί με το λεωφορείο;Τι νομίζετε;

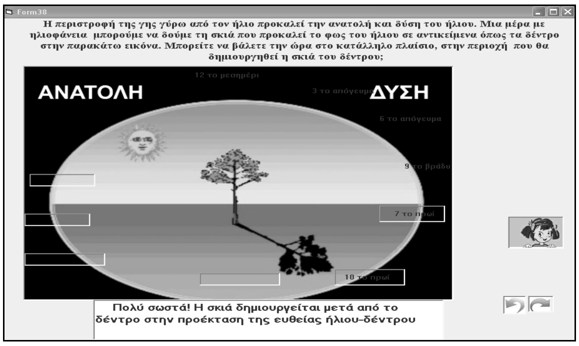
Πολύ σωστά! παρακολούθησε το βίντεο να το διαπιστώσεις

Θ2. Παρατηρήστε τη σκιά που δημιουργείται στο τοίχο τη στιγμή που προχωρά ο γεροντάκος και η γριούλα. Μπορεί να συμβαίνει αυτό;

Θ3. Τοποθετήστε τον ήλιο από την αριστερή μεριά του αγάλματος. Τι παρατηρείτε;

Θ4. Από ποιά μεριά του δέντρου θα ζωγραφίζατε τον ήλιο; Από την Α ή από τη Β; Τοποθετήστε τον εκεί που νομίζετε

Figure 4. Activities aiming to identify the objects that correspond to their shadows

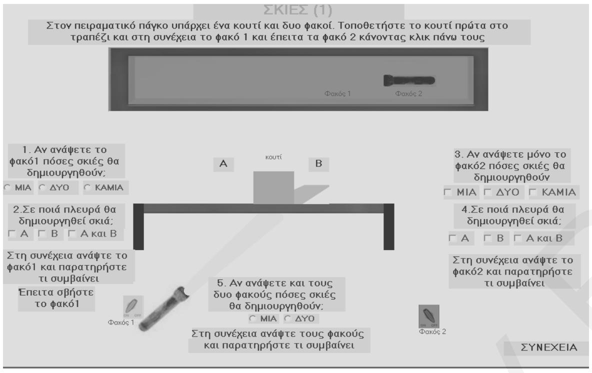


Η περιστροφή της γης γύρω από τον ήλιο προκαλεί την ανατολή και δύση του ήλιου. Μια μέρα με ηλιοφάνεια μπορούμε να δούμε τη σκιά που προκαλεί το φως του ήλιου σε αντικείμενα όπως τα δέντρο στην παρακάτω εικόνα. Μπορείτε να βάλετε την ώρα στο κατάλληλο πλαίσιο, στην περιοχή που θα δημιουργηθεί η σκιά του δέντρου;

Πολύ σωστά! Η σκιά δημιουργείται μετά από το δέντρο στην προέκταση της ευθείας ήλιου-δέντρου

Figure 5. Giving the right orientation of the object's shadow

The next screen shot allows students to test this very own hypothetical model ('run my-model', see Raghavan and Glaser, 1995), encouraging them to question their primitive ideas and help them adopt the scientific model. They can also activate an applet changing the distance of a light source from an object to investigate the way this changes the size of the shadow (Figure 7).



Στον πειραματικό πάγκο υπάρχει ένα κουτί και δυο φακοί. Τοποθετήστε το κουτί πρώτα στο τραπέζι και στη συνέχεια το φακό 1 και έπειτα τα φακό 2 κάνοντας κλικ πάνω τους

1. Αν ανάψετε το φακό1 πόσες σκιές θα δημιουργηθούν;
○ ΜΙΑ ○ ΔΥΟ ○ ΚΑΜΙΑ

2.Σε ποιά πλευρά θα δημιουργηθεί σκιά;
□ Α □ Β □ Α και Β

Στη συνέχεια ανάψτε το φακό1 και παρατηρήστε τι συμβαίνει

Έπειτα σβήστε το φακό1

3. Αν ανάψετε μόνο το φακό2 πόσες σκιές θα δημιουργηθούν
□ ΜΙΑ □ ΔΥΟ □ ΚΑΜΙΑ

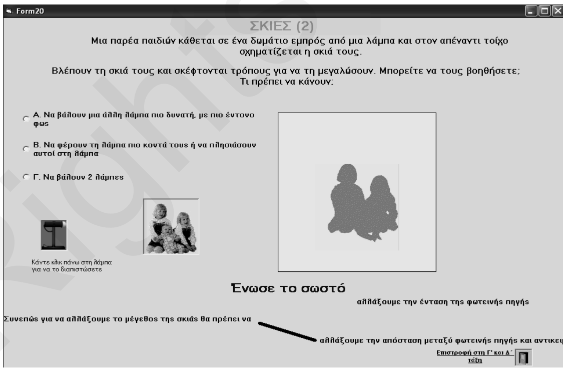
4.Σε ποιά πλευρά θα δημιουργηθεί σκιά;
□ Α □ Β □ Α και Β

Στη συνέχεια ανάψτε το φακό2 και παρατηρήστε τι συμβαίνει

5. Αν ανάψετε και τους δυο φακούς πόσες σκιές θα δημιουργηθούν;
○ ΜΙΑ ○ ΔΥΟ

Στη συνέχεια ανάψτε τους φακούς και παρατηρήστε τι συμβαίνει

Figure 6. Virtual experiments looking at the orientation of shadows



Μια παρέα παιδιών κάθεται σε ένα δωμάτιο εμπρός από μια λάμπα και στον απέναντι τοίχο σχηματίζεται η σκιά τους.

Βλέπουν τη σκιά τους και σκέφτονται τρόπους για να τη μεγαλώσουν. Μπορείτε να τους βοηθήσετε; Τι πρέπει να κάνουν;

○ Α. Να βάλουν μια άλλη λάμπα πιο δυνατή, με πιο έντονο φως

○ Β. Να φέρουν τη λάμπα πιο κοντά τους ή να πλησιάσουν αυτοί στη λάμπα

○ Γ. Να βάλουν 2 λάμπες

Κάντε κλικ πάνω στη λάμπα για να το διαπιστώσετε

Ένωσε το σωστό

Συνεπώς για να αλλάξουμε το μέγεθος της σκιάς θα πρέπει να

αλλάξουμε την ένταση της φωτεινής πηγής

αλλάξουμε την απόσταση μεταξύ φωτεινής πηγής και αντικει

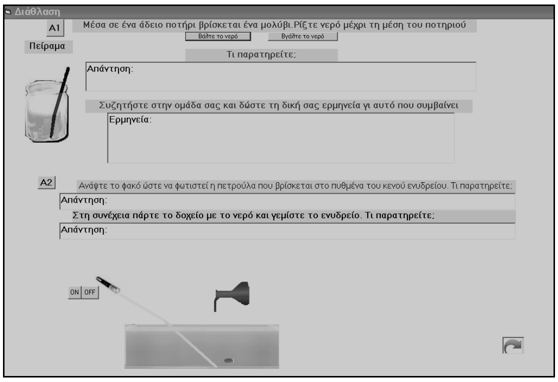
Figure 7. Activating the applet testingthe size of the shadow

Fourth category: Our initial research indicated that the majority of the 5th and 6th grade students believed that refraction was due to some property of objects themselves. When asked about the change in the appearance of a pencil when it is partly submerged in a glass of water, they attributed this to the shape of the pencil itself. 'Light-Life' includes the following features to help students confront and re-evaluate these initial conceptions. In the first instance students can formulate their own interpretation for the phenomenon of refraction as they can observe the change in the appearance of a pencil submerged in a glass of water. After that, they are engaged in a virtual experiment in which they fill a tank with water and see how the route taken by the light ray is changed as it comes in the water (Figure 8). Aiming to help students construct a suitable analogy, the next screen shot presents a hypothetical scenario in which they must choose the quickest route which a lifeguard in a swimming pool must take to reach an individual who is at risk of drowning (see Hewitt, 1997). They see that the lifeguard will reach the individual in distress quicker not by taking a direct route but by running further along the pool so that the distance he has to swim is kept to an absolute minimum, bearing in mind that the lifeguard can run faster than he can swim (Figure 8). After formulating their own hypotheses, the students can activate the scene to compare the time taken for the lifeguard to follow the two possible routes. They are also able to watch an applet on the internet (http://micro.magnet.fsu.edu/primer/java/particleorwave/refraction/index.html), where they can observe how the course of a light ray alters as it passes from a medium to a denser one. Finally, the students are prompted to draw their conclusions about the refraction phenomenon and reconsider the answers they gave to the question initially.

Fifth category: From our initial research it seemed that children aged 10 and 11 had no awareness of the directionality of light in sight processes, and instead of representing light they drew a geometrical connection between the eye and the viewed object. Also, when we asked students to show the direction of the light beam

Copy Rights

in the classical triangle (observer's eye, light lamp, object) on their worksheets, the emission model was the dominant model in their drawings. In order to cope with these students' alternative ideas the software 'Light-Life' provides animated graphic renditions, representations of vision, which depict the directionality of light. Bearing in mind that a single arrow is highly schematic and thus might not be representative enough of the idea of light transmission, multiple arrows were used to represent light emanating from a seen object, some of them reaching the eye. In a drawing, showing a child watching a flower, students were asked to predict the direction of light by choosing the correct arrows and then to test their hypothesis by activating the animation.



Διάθλαση

A1 Μέσα σε ένα άδειο ποτήρι βρίσκεται ένα μολύβι.Ρίξτε νερό μέχρι τη μέση του ποτηριού

Πείραμα

Τι παρατηρείτε;

Απάντηση:

Συζητήστε στην ομάδα σας και δώστε τη δική σας ερμηνεία γι αυτό που συμβαίνει

Ερμηνεία:

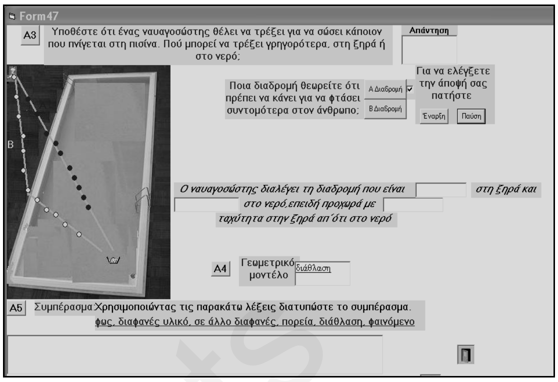
A2 Ανάψτε το φακό ώστε να φωτιστεί η πετρούλα που βρίσκεται στο πυθμένα του κενού ενυδρείου. Τι παρατηρείτε;

Απάντηση:

Στη συνέχεια πάρτε το δοχείο με το νερό και γεμίστε το ενυδρείο. Τι παρατηρείτε;

Απάντηση:

Figure 8. Observing the phenomenon of refraction



Form47

A3 Υποθέστε ότι ένας ναυαγοσώστης θέλει να τρέξει για να σώσει κάποιον που πνίγεται στη πισίνα. Πού μπορεί να τρέξει γρηγορότερα, στη ξηρά ή στο νερό;

Απάντηση

Ποια διαδρομή θεωρείτε ότι πρέπει να κάνει για να φτάσει συντομότερα στον άνθρωπο;

Για να ελέγξετε την άποψή σας πατήστε

Ο ναυαγοσώστης διαλέγει τη διαδρομή που είναι στη ξηρά και στο νερό,επειδή προχωρά με ταχύτητα στην ξηρά απ΄ότι στο νερό

A4 Γεωμετρικό μοντέλο: διάθλαση

A5 Συμπέρασμα Χρησιμοποιώντας τις παρακάτω λέξεις διατυπώστε το συμπέρασμα. φως, διαφανές υλικό, σε άλλο διαφανές, πορεία, διάθλαση, φαινόμενο

Figure 9. Hypothetical scenario with the lifeguard

Copy Rights

3. CONCLUSION

'Light-Life' is multimedia educational software that was developed as a tool for enhancing students' learning with understanding and teaching of geometrical Optics mainly in primary education. The initial research with 40 students using clinical interviews revealed their alternative conceptions (grouped in five categories), on which we were based to design this software. It comprises many characteristics, aiming to cope with students' alternative conceptions and to help them construct appropriate knowledge. The software's main characteristics are students' engagement in real problem solving activities, prediction and testing of hypotheses, creating and comparing their own models with the scientific ones, simulations of real life situations. In order to bridge the 'zone of proximal development' (Vygotsky, 1978), we provide scaffolding by referring to common everyday situations, providing challenging authentic activities requiring reflective thinking to construct a suitable analogy, and also providing students with opportunities to work in collaborative groups. Students do not process the full complexity of the problem from the very beginning, but face a simpler version of it. So, scaffolding takes place and students achieve better learning outcomes. Basic regularities provided by the software allow students to recode the information pertaining to the complex problem. We thereby aim to lead students to successful reformulation, and explanations of daily problems.

We made a preliminary research on the evaluation of the software by using it in a primary classroom with 23 6th grade students. The results of this initial study are very encouraging, as the students found the software easy to use and achieved good learning results. The next step is to use of the software in more than one class and over a range of different grades. This study will allow us to test the effectiveness of the software using a larger sample of students and compare those outcomes with the outcomes achieved using more traditional teaching methods. Such an evaluation should be of significant assistance to teachers who seek to improve their teaching, to designers of educational software, and to those looking to improve both the teaching materials and methods for these areas of education.

REFERENCES

Andersson, B. and Karrqvist, C., 1983. How Swedish pupils, aged 12–15 years, understand light and its properties. *European Journal of Science Education,* Vol. 5, No 4, pp 387–402.

Bendall, S. et al., 1993. Prospective elementary school teachers' prior knowledge about light. *Journal of Research in Science Teaching*, Vol. 30, pp 1169-1187.

Bransford, J. et al., 2000. *How people learn: brain, mind, experience, and school*, Washington, D.C., National Academy Press.

Driver, R., 1989. Students' conceptions and the learning of science. *International Journal of. Science Education*, Vol. 11, No .5, pp 481–490.

Driver, R. et al., 2000. Making *sense of secondary science: research into children's ideas*. Routledge, London

Driver, R. and Oldham, V. 1986. A constructivist approach to curriculum development in science**.** *Studies in Science Education,* Vol. 13, pp. 105–122.

Feher, E. and Rice, K., 1988. Shadows and anti-images: Children's conceptions of light and vision. *Science Education*, Vol. 72, No. 5, pp. 637-649.

Fetherstonhaugh, T. et al., 1987. Student alternative conceptions about light: A comparative study of prevalent views found in Western Australia, France, New Zealand, Sweden and the United States. *Research in Science Education,* Vol. 17, pp. 156-164.

Fetherstonhaugh, T. and Treagust D.F., 1992. Students' understanding of light and its properties: teaching to engender conceptual change. *Science Education*, Vol. 76, No 6, pp. 653-672.

Feynman, et al., 1964. *The Feynman lectures on physics*. Reading, MA: Addison-Wesley.

Galili, I. and Hazan, A., 2000. Learners' knowledge in optics: interpretation, structure and analysis. *International Journal of Science Education*, Vol. 22, No. 1 pp. 57–88.

Gregory, R.L., 1979. *Eye and Brain.* Princeton. NJ: Princeton University Press.

Guesne, E., 1985. Light. In R. Driver, E. Guesne & A. Tiberghien (Ed.) *Children's Ideas in Science* (pp. 10-32) Philadelphia: Open University Press.

Hewitt, P., 1997. *Physics Concepts* (1) (E. Sifaki, trad.). Heraklion: University Publications of Crete (in Greek).

Hosson, C. and Kaminski W., 2002. Les yeux des enfants sont-ils des 'porte-lumière'? *Bulletin de l'union des physiciens* Vol. 840, pp.143–160.

Jonassen, D.H., 1993. *Computers in the Classroom: Mindtools for Critical Thinking*, Englewood Cliffs, New Jersey, Prentice Hall.

Jonassen, D.H., 1999. Designing constructivist learning environments. In: Reigeluth CM (Ed.) *Instructional-Design Theories and Models*, vol II pp. 215–239. Lawrence Erlbaum Associates, New Jersey.

Langley, D. et al., 1997. Light propagation and visual patterns: pre-instruction learners' conceptions. *Journal of Research in Science Teaching,* Vol. 34, No. 4, pp. 399-424.

Osborne, J.F., 1996. Beyond constructivism**.** *Science Education* Vol. 80, pp.53–82.

Osborne, J. et al., 1993. Young children's (7–11) ideas about light and their development. *International Journal of. Science Education,* Vol 15. No. 1, pp 83–93.

Piaget, J. (1928). *The Child's Conception of the World.* London: Routledge and Kegan Paul.

Raghavan, K. and Glaser, R., 1995. Model-based analysis and reasoning in science: The MARS curriculum. *Science Education*, Vol. 79, pp. 37–61.

Ravanis, K., 1999. Représentations des élèves de l'école maternelle: le concept de lumière. *International Journal of Early Childhood* Vol. 31, No. 1, pp. 48–53.

Ravanis, K. et al., 2002. Social marking and conceptual change: the conception of light for ten-year old children. *Journal of Science Education,* Vol 3, No 1, pp. 15-18.

Ronchi, V., 1970. *The Nature of Light,* Cambridge, MA: Harvard University Press.

Selley, N.F., 1996. Children's ideas on light and vision. *International Journal of Science Education,* Vol. 18, No. 6, pp. 713–723.

Solomon, J., 1994. The rise and fall of constructivism. *Studies in Science Education* Vol. 23, No. 1, pp. 1-19.

Solomonidou, C., 2006. *New trends in educational technology. Constructivism and new learning environments*. Athens: Metaihmio editions (in Greek).

Tekos, G. et al., 2008, Teaching light reflection and diffusion using constructivist digital tools and methods in Greek primary school. *ED-MEDIA* AACE, Luka, J. and Weippl E. (Ed), Vienna, Austria, pp.138-139.

Tekos, G. and Solomonidou, C., 2009. Constructivist learning and teaching of Optics concepts using ICT tools in Greek primary school: A pilot study. *Journal of Science Education and Technology,* Vol. 18, No. 3, pp. 415-428.

Tiberghien, A. et al., 1980. Conceptions de la lumière chez l'enfant de 10–12 ans. *Revue Française de Pédagogie,* Vol. 50, 24-41.

Vygotsky, L.S., 1978. *Mind in society: the development of higher psychological processes*. Cambridge: Harvard University Press.

Watts, M., 1985. Student conceptions of light: a case study. *Physics Education,* Vol. 20, pp. 183-187.

Copy Rights

A METHODOLOGY FOR ENGINEERING REAL-TIME INTERACTIVE MULTIMEDIA APPLICATIONS ON SERVICE ORIENTED INFRASTRUCTURES

Dimosthenis Kyriazis[1], Ralf Einhorn[2], Lars Fürst[2], Michael Braitmaier[3], Dominik Lamp[3], Kleopatra Konstanteli[1], George Kousiouris[1], Andreas Menychtas[1], Eduardo Oliveros[4],
Neil Loughran[5] and Bassem Nasser[6]
[1]*National Technical University of Athens*
[2]*Tixeltec*
[3]*University of Stuttgart*
[4]*Telefonica Investigation e Disarollo*
[5]*SINTEF*
[6]*University of Southampton IT Innovation Centre*

Copy Rights

ABSTRACT

Future Internet applications raise the need for environments that can facilitate real-time and interactivity without major modifications in the application domain. Such environments should be able to efficiently adapt resource provisioning to the dynamic demands of the applications, the majority of which tends to be real-time and interactive. In principle, all applications are suitable to be executed in service oriented environments as they are, without any kind of adaptation or modification. Nevertheless, the concrete characteristics of emerging infrastructures, mainly focused on guaranteeing the offered quality of service, require specific steps to be performed by the application developers and adopters in order to exploit the maximum of the infrastructure offerings. In this paper, we present a methodology for creating or adapting real-time interactive multimedia applications for service oriented infrastructures.

KEYWORDS

Service-oriented Infrastructures, Cloud computing, Quality of Service, Real-time

1. INTRODUCTION

Service Oriented Architectures (SOAs) [1] refer to a specific architectural paradigm that emphasizes implementation of components as modular services that can be discovered and used by clients. Infrastructures based on the SOA principles are called Service Oriented Infrastructures (SOIs). Through the agility, scalability, elasticity, rapid self-service provisioning and virtualization of hardware, Service Oriented Architecture principles are reflected into Clouds, which provide the ability to efficiently adapt resource provisioning to the dynamic demands of Internet users. Many architectural paradigms from distributed computing such as service-oriented infrastructures, Grids and virtualisation are incorporated into Clouds. There are three main classes in the cloud services stack which are generally agreed upon [2]:

- *Infrastructure as a Service* (IaaS), which refers to the provision of 'raw' machines (servers, storage, networking and other devices) on which the service consumers deploy their own software (usually as virtual machine images).
- *Platform as a Service* (PaaS), which refers to the provision of a development platform and environment providing services and storage, hosted in the cloud.
- *Software as a Service* (SaaS), which refers to the provision of an application as a service over the Internet or distributed environment.

In this paper we focus on the SaaS class, presenting a methodology for adapting real-time interactive multimedia applications for cloud-based service oriented infrastructures. Before we continue, let us clarify the term "real-time". Traditionally, 'real time' refers to hard real-time systems, where even a single violation of the desired timing behaviour is not acceptable, for example because it leads to total failure, possibly causing loss of human lives. However, there is also a wide range of applications that also have stringent timing and performance needs, but for which some deviations in Quality of Service (QoS) are acceptable, provided these are well understood and carefully managed. These are soft real-time applications and include a broad class of interactive and collaborative tools and environments, including concurrent design and visualization in the engineering sector, media production in the creative industries, and multi-user virtual environments in education and gaming. In particular, we focus on interactive soft real time applications where one or more users interact with the application and with each other.

Soft real-time applications are traditionally developed without any real-time methodology or run-time support from the infrastructure on which they run. The result is that either expensive and dedicated hardware has to be purchased to ensure good interactivity levels and performance, or that general-purpose resources are used as a compromise (e.g. commodity operating systems and Internet networking) with no way to guarantee or control the behaviour of the application as a result. In this paper, we present a methodology, being developed in the European Commission supported IRMOS project [3], for application developers describing how to use specific tools during various phases of the application development process. The outcome of this process is soft real-time applications that can be executed on Service Oriented Infrastructures (SOIs) with guaranteed quality levels. The proposed adaptation enables the PaaS providers to utilize techniques for modelling, predicting, provisioning and monitoring resource and QoS requirements commitments and thus allowing real-time interactive multimedia applications to be executed on SOIs.

The aforementioned applications consist of *Application Components* (ACs) and are described by an Application Description (AD) that includes their functional parameters. The ACs are software components actually providing functionality and can be: a) *Application Service Components* (ASCs) that run autonomously and are deployed inside the virtualized infrastructure, e.g. an image renderer, b) *External ASCs* (EASCs) – same as ASCs but run outside the virtualized infrastructure because they offer non-standard, “non-virtualizable” functionality, e.g. a GPU processing node, c) *Application Client Components* (ACCs) – the software used by an end-user (client) to access the application – always running outside the virtualized infrastructure. The ACs are described by a document containing everything that is needed (e.g. resource needs and functional parameters) to run the ASC on the SOI. This is called *Application Service Component Description* (ASCD).

The remainder of the paper is structured as follows: Section 2 gives an overview on characteristics of potential applications and first generic (i.e. independent of the SOI) steps to be done for preparation, while Section 3 focuses on the steps needed for integration and adaptation the applications to a real-time aware cloud-based platform (e.g. IRMOS) and available tools helping the developer on this task. The functional and descriptive interfaces used in these processes are described in Section 4. The paper concludes with a discussion on future research and potentials for the current study.

Copy Rights

2. THE FOUNDATION: AN APPLICATION

Basically there are two possibilities to create an application that will run on a service oriented infrastructure:

- Pre-existing applications can be adapted to run on it (“adaptation”).
- New applications developed tailored to the infrastructure (“green field development”).

Instead of running on dedicated physical hosts, the service parts of the applications run on Virtual Machine Units (VMUs) within a virtualized infrastructure provided by the IaaS provider, an example of which is Intelligent Service Oriented Network Infrastructure – ISONI [4].

In principle, all server applications are suitable to be executed in the cloud-based environment as they are, without any kind of adaptation or modification. But the concrete characteristics of platforms that facilitate real-time and interactivity, make applications that require live interactions among their users or which are infrastructure demanding (in terms of CPU, network or storage usage) more appropriate for such platforms.

Applications that require scaling resources or have changing use patterns over time require adaptation of the infrastructure according to the specific usage periods and reservation of more or less resources depending on the load in each moment. In other cases, there are applications that not only have important requirements in terms of resource usage and depend on reliable resources but also need QoS guarantees. IRMOS platform is considered to be suitable for the aforementioned applications by allowing both for interactivity but also for

QoS guarantees across a virtualized infrastructure. The design provides certain advantages in relation to different aspects related with security, as for instance: the isolation during execution from other applications (that caused by misbehaviour or malicious software) could try to monopolise all resources in the machine, damaging other applications executing over the same physical machine. There is also a complete isolation at network level that prevents applications (like sniffers) running inside the environment to access data of other surrounding applications.

Following, we describe the steps that are required in order to prepare an application to be executed on a SOI. These steps are the following:

- *Step 1 - Application Components Creation*: The application has to be split into application components (ACs) at least separating the user interface part (ACC) from the computing part(s) (ASC(s)). For partitioning think of each component running on a different physical host. Think of this separation as a transformation of the application from a monolithic one towards a distributed program that is able to run on distributed systems [5], [6]. How the application is separated depends on the specific application logic and what should be achieved. While it may be convenient for one application to be split in a large number of modules as the module specific tasks have a high degree of reusability (think of the topic of componentized development and service-oriented applications), it might be different for another application that has specific subtasks which have to be componentized and put on, e.g. highly demanding computational resources. The conclusion is that "componentizing" your application is the way to utilize the SOI by making use of scalable QoS-guaranteed resources in an "on-demand"-fashion.
- *Step 2 - Components Interfaces Creation*: Components must be equipped with interfaces to the SOI:
  - ASCs need a run-time interface to be configured, controlled and monitored by the framework. The purpose of this interface is to act as a mediator between the AC and the infrastructure framework services, represented on the application component's node by a framework service that gathers the aforementioned information. However the amount of information might vary according to the information made available by the AC. The more information an AC reveals to the SOI the better the infrastructure can deal with QoS issues for that component.
  - ACC need to be started with specific parameters required for accessing the application service running on the SOI. The configuration information provides the necessary details to allow the ACC to access the various components (ASCs) on the infrastructure that need to have direct connections to an ACC.
- *Step 3 - Components Packaging*: ASCs must be packaged for deployment.
- *Step 4 - Components Description*: ASCs must be described, especially resource-wise.
- *Step 5 - Application Description*: The application (as a whole) must be described.

The latter steps (Steps 4 and 5) are facilitated through various tools and modular approaches e.g. for wrapping ACs.

The following figure (Figure 1) shows a typical service oriented application. The application runs distributed on different hosts, while application related communication and data transfer like passing image data from one component to another is done independently from the platform (however its parameters have to be described for resource reservation). Framework related communication e.g. for controlling and monitoring ASCs is limited to the minimum (for having as little impact as possible on the actual application).

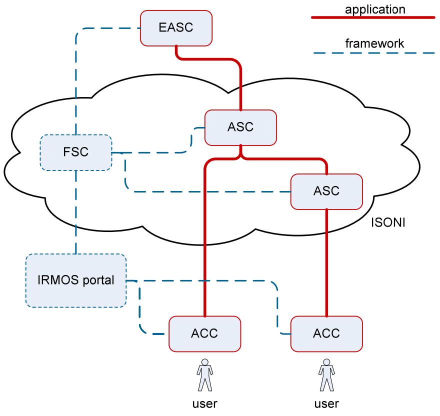


Figure 1. Sample application

Copy Rights

3. FROM APPLICATION COMPONENTS TO AN APPLICATION

As described in Section 2, a service oriented application is based on application components (ACs). Figure 2 depicts the phases of an AC. This section goes through the phases relevant for the application component developer as well as the application designer (the two upper boxes). However processes happening during the "use" phase like ASC configuration and execution have to be taken into account for creating ASCs.

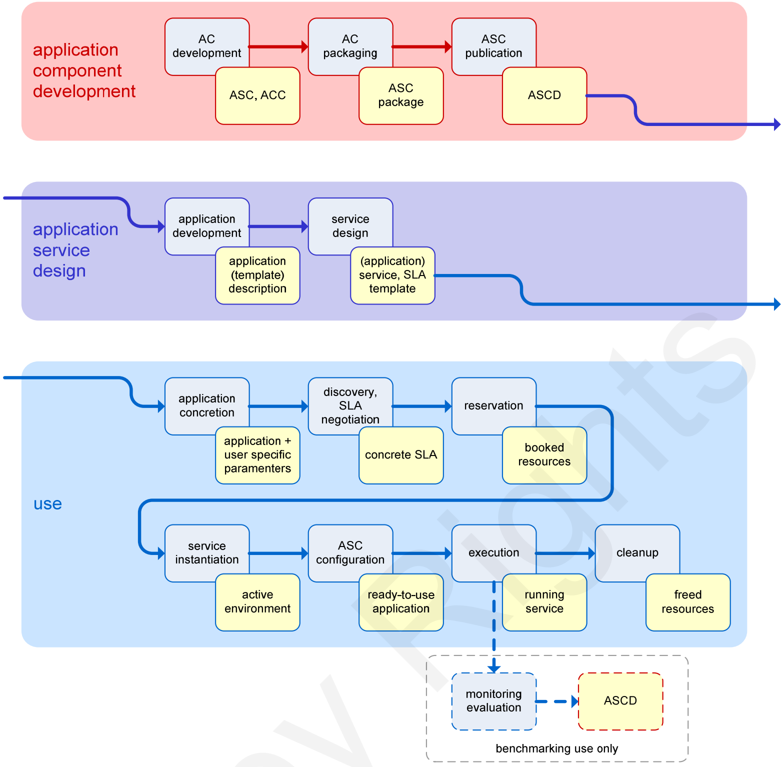


Figure 2. Application phases

Copy Rights

3.1 AC Development & Packaging

The first step in the process of adapting an application to run on a service oriented environment refers to the development of the application components. Ideally the created ACs may also be used in a non-SOI. Communication between the ACs (including control and pay load data transfer) must be implemented – if not already in place. Splitting up a monolithic application into components can also be realized by keeping the application core as it is but rather adding functionality (e.g. interfaces) which enables the core to behave as a specific component with dedicated functionality. Hence the same binary can be used e.g. as a GUI as well as a service part.

Like any other software ACs have to be packaged for deployment.

- EASCs are ACCs deployed independent of the SOI. Therefore the packages may be of any format and are not further discussed here.
- ASCs are to be deployed in the SOI – in defined package format(s).

The ASC to be deployed inside the VMUs, usually has to be provided as a standard .rpm package for UNIX operating systems. Particularly, the dependencies have to be specified completely. The standard mechanisms for recursive dependencies apply. All packages that are included for example in the standard Fedora repository can be specified as dependency and are automatically installed.

3.2 ASC Publication

ASCs must be published to the framework which includes the following two actions:

1. Publish a description of the ASC to the PaaS domain. This task implies that the ASCD has already been created. This ASCD is then uploaded to the ASC repository, a dedicated repository for storing data related to the ASCs. Note that this procedure only applies to ASCs, since ACCs are deployed separately (i.e. independent form the framework). For EASCs an ASCD is needed as well, but deployment takes place framework independent.
2. The ASC publication process also includes the publication of the corresponding ASC binary to an ASC repository. Among several other items the ASCD also contains a link to the binary ASC package.

The above two processes take place at the same step as depicted in Steps 1 to 3 in the publication sequence diagram (Figure 4). The Application Provider uploads a directory of predefined structure that includes both the ASCD as well as the binary, therefore the link to the binary ASC package that each ASCD carries, is actually a relative path pointing to the location of the binary inside its directory.

These steps are repeated for each ASC and each time the Application Provider obtains a link to the specific location where the information about each ASC is stored in the ASC repository.

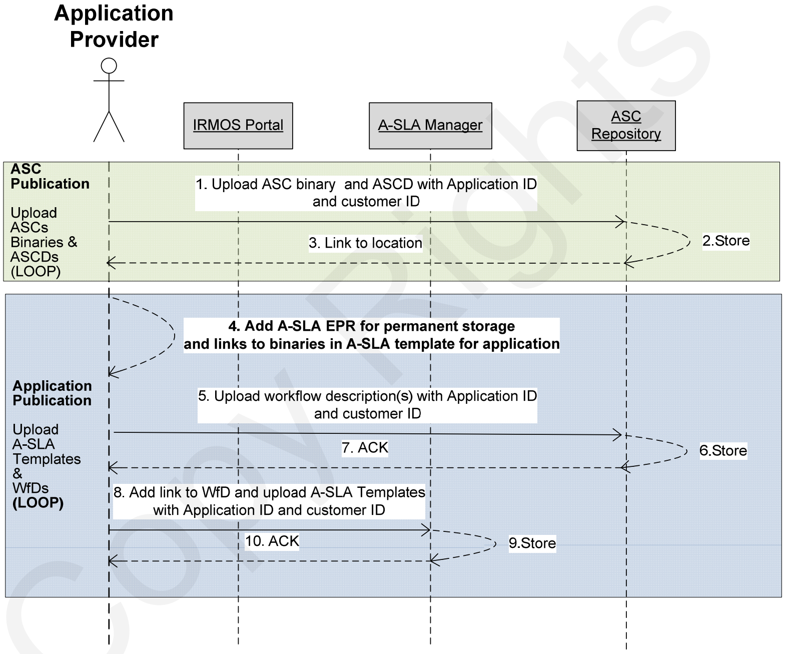

Application Provider
IRMOS Portal
A-SLA Manager
ASC Repository
ASC Publication
Upload ASCs Binaries & ASCDs (LOOP)
1. Upload ASC binary and ASCD with Application ID and customer ID
2.Store
3. Link to location
4. Add A-SLA EPR for permanent storage and links to binaries in A-SLA template for application
Application Publication
Upload A-SLA Templates & WfDs (LOOP)
5. Upload workflow description(s) with Application ID and customer ID
6.Store
7. ACK
8. Add link to WfD and upload A-SLA Templates with Application ID and customer ID
9.Store
10. ACK
Copy Rights

Figure 4. ASC and application publication sequence

3.3 Application Development

When creating an application the developer uses a service modeling environment (e.g. Papyrus [7]) that contains a profile for modelling ASCs. As there is a lot of detail involved in this we will only describe the process from a high level point of view. It should be noted that while we refer to the Papyrus tool in developing applications the approach is generic and thus the profile and process can be utilised in any UML design tool which supports the UML2 meta-model.

Similar to single components the entire application(s) derived from the components needs to be described. This means that components are selected and virtually interconnected. Several values for high level parameters may be pre-selected or its ranges may be restricted. However an application description (AD) remains a template where several parameter values can be selected by the user. As applications are derived from ASCs there parameters (the ones visible from outside) are also taken from the corresponding ASCDs.

There are several elements involved in creating an application:

3. Creation of ASC descriptions (using UML classes)
4. Creation of instances of these descriptions (using UML class instance specification)
5. Creation of links between instances (using UML composite structure diagram)
6. Creation of a workflow which describes the interaction between the ASC instances (using UML activity diagram)
7. Creation of an instance of the application (this is essentially an instance of all the above)

In the following sub-sections we will briefly cover these different elements.

3.3.1 ASC Descriptions

An ASCD can be described as the development of a set of properties which are specific to the domain in which the ASCD is to be used. For example in the film domain the developer may wish to describe parameters relating to video, production schedules and so on, while in another domain the set of parameters can be entirely different. However, each ASCD also includes a set of common parameters which are part of the ASCD profile (e.g. benchmarking properties, etc.). Therefore, the developer will typically set these values with concrete values in this description using the properties window. The descriptions are developed using UML class diagrams with the UML4ASCD profile. The following figure (Figure 5) illustrates a typical ASCD for a SimpleImageReszier application. It simply defines a class which has the ASCD profile assigned to it. Within an ASCD a number of properties are described which are also stereotyped with elements from the ASCD4UML profile (e.g. <<aSCParameter>>, <<benchmark>>, etc.).

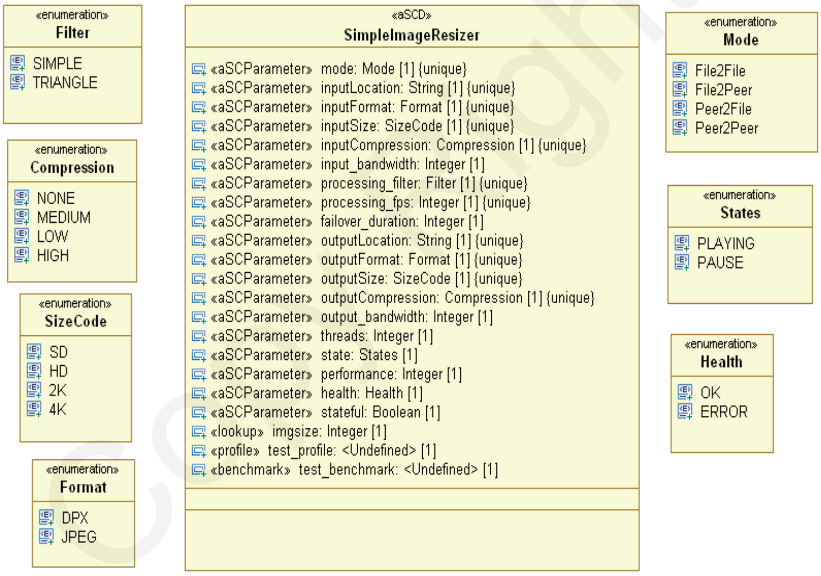

«enumeration»
Filter
SIMPLE
TRIANGLE
«enumeration»
Compression
NONE
MEDIUM
LOW
HIGH
«enumeration»
SizeCode
SD
HD
2K
4K
«enumeration»
Format
DPX
JPEG
«aSCD»
SimpleImageResizer
«aSCParameter» mode: Mode [1] {unique}
«aSCParameter» inputLocation: String [1] {unique}
«aSCParameter» inputFormat: Format [1] {unique}
«aSCParameter» inputSize: SizeCode [1] {unique}
«aSCParameter» inputCompression: Compression [1] {unique}
«aSCParameter» input_bandwidth: Integer [1]
«aSCParameter» processing_filter: Filter [1] {unique}
«aSCParameter» processing_fps: Integer [1] {unique}
«aSCParameter» failover_duration: Integer [1]
«aSCParameter» outputLocation: String [1] {unique}
«aSCParameter» outputFormat: Format [1] {unique}
«aSCParameter» outputSize: SizeCode [1] {unique}
«aSCParameter» outputCompression: Compression [1] {unique}
«aSCParameter» output_bandwidth: Integer [1]
«aSCParameter» threads: Integer [1]
«aSCParameter» state: States [1]
«aSCParameter» performance: Integer [1]
«aSCParameter» health: Health [1]
«aSCParameter» stateful: Boolean [1]
«lookup» imgsize: Integer [1]
«profile» test_profile: <Undefined> [1]
«benchmark» test_benchmark: <Undefined> [1]
«enumeration»
Mode
File2File
File2Peer
Peer2File
Peer2Peer
«enumeration»
States
PLAYING
PAUSE
«enumeration»
Health
OK
ERROR
Copyrights

Figure 5. Example ASC description

3.3.2 ASCD Instances

ASCD instances are essentially developed using UML class instance specification diagrams. Using these diagrams it is possible to develop instances which are typed based on the names of the ASCD descriptions. Figure 6 depicts a class instance which uses the aforementioned ASCD description. The instance simply provides a concrete value to the mode property. For purpose of brevity we only show one property with a value although the instance would typically contain several properties and assigned values.

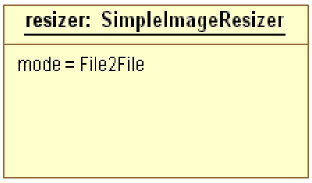


Figure 6. Example of ASC instance

An application description is typically made up of several interconnecting ASCD instances. In order to model this we use the UML composite structure diagram. Composite structure diagrams are used for modelling the static structure of the application, i.e. the components involved and how they are connected. As such, Figure 7 illustrates the previous resizer instance of SimpleImageResizer connecting with other ASCDs.

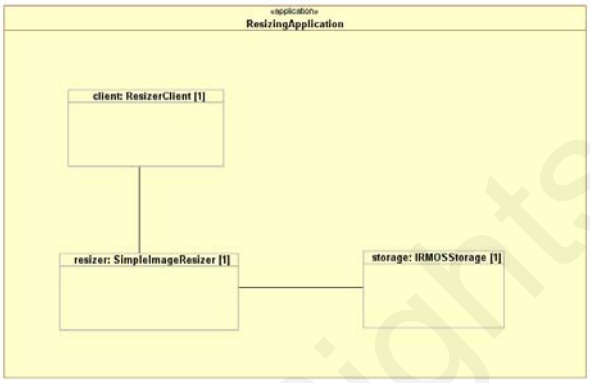

«application»
ResizingApplication
client: ResizerClient [1]
resizer: SimpleImageResizer [1]
storage: IRMOSStorage [1]

Figure 7. Example of linking ASC instances

3.3.3 Application Instance

The previous sub-sections illustrated how to define an application composed of components. This gives the structure and the types of the application, but it does not provide the actual configurations for the different ASCDs. As mentioned these configurations are in the ASC instances. So the task is to fill an instance of the application class with the ASCD instances. To do this we use the UML composite structure diagram and draw some 'slots' then assign them with ASCD instances as shown in Figure 8.

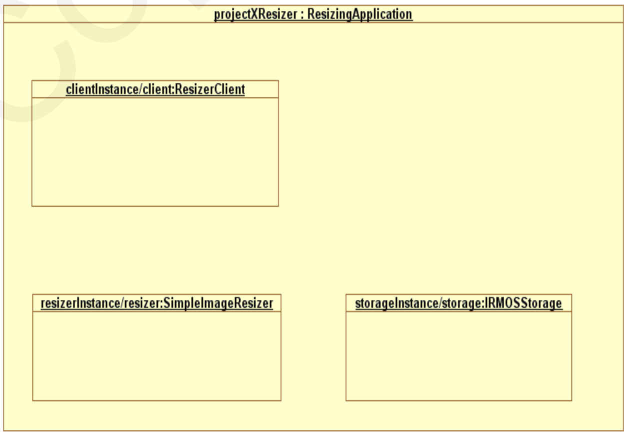

projectXResizer : ResizingApplication
clientInstance/client:ResizerClient
resizerInstance/resizer:SimpleImageResizer
storageInstance/storage:IRMOSStorage

Figure 8. Application instance example

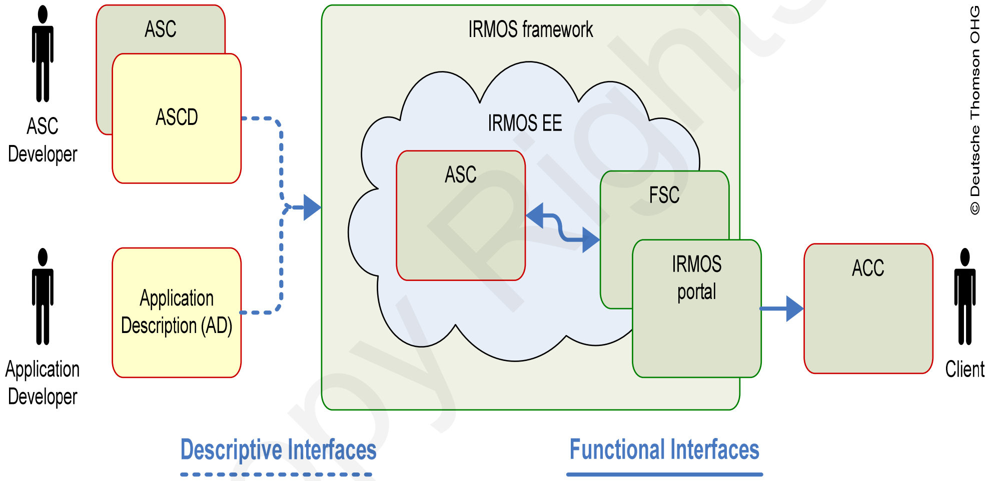
Copy Rights

3.4 Application Publication

Having published the ASCs (Section 3.2) and created the A-SLA templates that correspond to different ways of using the same application (different workflow and/or different level of QoS), the Application (SaaS) Provider publishes these A-SLA templates to the PaaS Provider domain by uploading the corresponding descriptive files to the dedicated repository (via the A-SLA Manager), as depicted at steps 4 to 10 in the publication sequence diagram (Figure 4). It should be stressed at this point that each A-SLA template builds heavily on the ASCDs and also includes a link to the workflow description of the application it represents as well as a reference to the permanent storage that is shared by all customers of the application. In order to be coherent with what is described in the ASC publication, the workflow description could also be stored inside the application's corresponding folder in the ASC repository.

4. THE INTERFACES

Although the interfaces are an essential part of the ASC development (i.e. they are used during development) they and their usage are described separately in order to keep Section 3 more compact. The following figure (Figure 9) shows the different AC-relevant interfaces both descriptive and functional.



Copy Rights

Figure 9. Interfaces

An ASCD thoroughly describes an ASC and represents the only 'off-line' interface between the application and the SOI. Hence it is an essential document containing the entire set of information letting the environment use the ASC regarding resource reservation, configuration, control, monitoring, benchmarking and modelling. An ASCD is built using a UML design tool and the associated ASCD4UML profile (Figure 10). For simplicity we have chosen to use this profile within the Papyrus UML design tool due to its open source nature. However, as UML is an adopted standard there is nothing to stop the developer using any UML design tool of their choice along with the previously mentioned ASCD4UML profile. The profile is purposely designed to be easy to use within any UML design tool such as Papyrus (which is the tool used in this guide). The profile consists of a number of different stereotypes, as illustrated in Table 1.

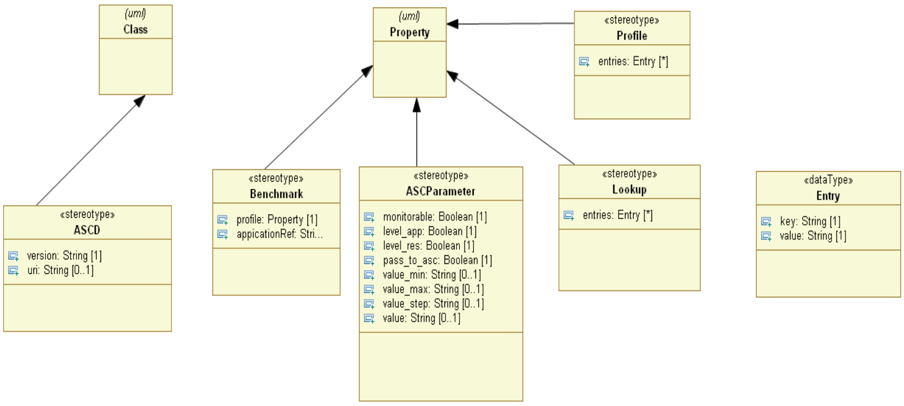


Figure 10. UML profile for ASCD

Table 1. Description of UML4ASCD stereotypes

Stereotype	Description
<<aSCD>>	Identifies a class as being an ASCD
<<aSCParameter>>	Identifies a property as being an ASC parameter within an ASCD
<<lookup>>	Identifies a property as a lookup table within an ACSD
<<profile>>	Identifies a property as profile within and ASCD
<<benchmark>>	Identifies a property as a benchmark within an ASCD

The main idea is to allow different stereotypes to be applied to specific properties we want to define in our intended application. When defining an ASCD, we can also provide values to different properties which are contained within the profile elements.

4.1 Defining Resources on High and Low Level

One of the main purposes of the ASCD is to let the PaaS provider generate a description of the required resources for executing an ASC on the virtualized infrastructure. The aforementioned description requires values for low level parameters (e.g. “CPU frequency”) completely independent of a specific application. Therefore application (“high”) level parameters values (e.g. “frames per second”) must be translated (“mapped”) to low level ones.

The ASC developer has to select which low level parameters must be provided for her/his ASC to run as well as the high level parameter for configuration of the ASC – which also might have an impact on performance.

Basically two tables of parameters have to be created.

- The high level parameter table contains all parameter inputs on application level. These parameters must contain the range of the possible values of the parameter or an enumeration of all possible values that these parameters may take if they have discrete values. It is best that these values are numerically represented, e.g. possible resolutions which have discrete allowed values (800x600, 1024x768 etc.) can be added as numbers of pixels (480000 pixels, 786432 pixels).
- The low level parameter table contains all parameters needed for executing the ASC on the virtualized infrastructure without any actual values (as these are calculated by the PaaS provider).

The ASC developer must specify what inputs he wants to map with what outputs. A flag in the ASCD can be used for this purpose. These outputs can then be used in the ASLA for determining the QoS levels of the ASC or the application in general.

Copy Rights

4.2 Defining Functional Parameters

Besides parameters relevant for resource dimensionality there are a couple of parameters necessary for the ASC to run, e.g. an IP address of a peer node. Some parameters may not be relevant to the FS at all but just have to be passed (e.g. an operation mode of the ASC selected by the user). Others have impact on calculating low level parameters and will also be used (indirectly) for the description of the required resources. Others (IP addresses) will directly be used for both, description of the resources and ASC configuration.

Hence functional parameters can be both – high level or low level. However their values are simply defined without the need of processing them for resource estimation.

4.3 Providing Information for Benchmarking

In order to let the FS determine resource mappings (i.e. low level resource values for specific high level application parameters) an ASC needs to be benchmarked. Benchmarking requires a dedicated set of parameter (profiles) as well as data for input and output. Benchmarking is usually conducted as a special case of the normal execution phase. The reason for this is twofold. First of all, extra delays that are inserted by the virtualization layer in the infrastructure and by the rest of the PaaS services need to be measured. By benchmarking on realistic infrastructures the data set will be more precise and so will be the estimations. Furthermore, the Framework Services do not have the necessary infrastructure to perform benchmarking, from physical machines to specialized monitoring services, according real-time schedulers, intelligent network links etc., since this is not their purpose in the overall architecture.

In order for this to be implemented, the ASC needs to follow all the steps of a normal application design (like workflow description and modelling), in order to be able to be executed as a standalone application, for the aforementioned reasons. This “benchmarking workflow” will contain the specific ASC plus several other elements that are required for it to be executed.

The ASC developer must also provide the high level values for which to benchmark and the according input files (e.g. he must specify that he will run the video encoding ASC with a video of 25 fps, and provide an according video). These values for which to benchmark can be inserted into the ASCD or in a better fashion the ASC developer can edit a specific purpose ASLA for benchmarking, so that he can also control the cost of the benchmarking phase.

4.4 Providing Information for Modelling

In order to let the PaaS provider improve estimations for resources (and consequently their exploitation) a model of the ASC has to be provided. Application modelling aims at understanding the application’s behaviour in terms of key performance indicators, i.e. workload completion time, mean time to failure, mean time for recovery from failure, availability.

The application model refers to one or more ASCs, which are then combined to determine the behaviour of an application as a whole. The application developer has to specify this combination which is referred to as Application Description (AD).

The application performance estimation builds on the application model to estimate required resources to be allocated. This requires the following information to be available in the ASC model:

1. Workload Features: these are the characteristics of the workload that the customer is allowed to submit e.g. average video length, video format.
2. ASC Interrupt Events: specification of the interactions with the application that this customer is allowed to do e.g. stop/pause the application.
3. ASC FSM: This includes the different states and transitions (in terms of probability values) that affect the ASC execution. It also includes resource failure models that may affect the ASC runtime e.g. link failure, bandwidth drop below certain limit, etc.
4. ASC Normal Operation Time Estimator: this is the tool (e.g. neural-network) to be used for estimating the ASC uninterrupted fault-free completion time. This requires selecting a representative benchmark suite tests with which the ASC is benchmarked.

Copy Rights

5. CONCLUSION

Current approaches on Service Oriented Architectures focus on designing and implementing a rich set of services to efficiently operate, manage and reconfigure computing, storage and network resources under real-time conditions, providing to end users and to the associated applications the appropriate and required level of QoS. All Platform and Infrastructure capabilities are offered as on-demand services, although the architecture of the media applications varies from traditional n-tier enterprise applications to service-oriented workflows. Thus emerging cloud-based platforms and service oriented infrastructures face the challenge of providing QoS guarantees by proper real-time CPU scheduling [8] in order to facilitate real-time and interactivity as requested by Future Internet Applications.

In this paper we presented a methodology on how to engineer real-time interactive multimedia applications on service oriented infrastructures. The methodology describes the steps needed for integration and adaptation of the applications as well as their functional and descriptive interfaces to SOIs. Given the multi-tenanted development tools provided by PaaS providers (e.g. Facebook), such methodologies are considered essential not only for a limited number of application developers but also for consumers who also tend to become application developers.

ACKNOWLEDGEMENT

This work has been supported by the IRMOS project and has been partly funded by the European Commission’s IST activity of the 7th Framework Programme under contract number 214777.

REFERENCES

[1] T. Erl, “Service-oriented Architecture: Concepts, Technology, and Design”, Upper Saddle River: Prentice Hall PTR, ISBN 0-13-185858-0, 2005.

[2] The NIST Definition of Cloud Computing, Peter Mell and Tim Grance, Version 15, http://csrc.nist.gov/groups/SNS/cloud-computing, 2009

[3] The IRMOS Project, www.irmosproject.eu

[4] Whitepaper, “Intelligent Service Oriented Network Infrastructure Whitepaper”, 2009.

[5] Ghosh, Sukumar (2007), Distributed Systems – An Algorithmic Approach, Chapman & Hall/CRC, ISBN 978-1-58488-564-1

[6] Lynch, Nancy A. (1996), Distributed Algorithms, Morgan Kaufmann, ISBN 1-55860-348-4

[7] Papyrus is an open source graphical UML modelling tool; see http://www.papyrusuml.org. A bundle containing Papyrus as well as IRMOS specific extensions has been created.

[8] Fabio Checconi, Tommaso Cucinotta, Dario Faggioli, Giuseppe Lipari, "Hierarchical Multiprocessor CPU Reservations for the Linux Kernel," in Proceedings of the 5th International Workshop on Operating Systems Platforms for Embedded Real-Time Applications (OSPERT 2009), Dublin, Ireland, June 2009

Copy Rights

A SELF LEARNING CONTEXT-AWARE DOMOTICS SYSTEM TO AUTOMATE USER ACTIONS

Niels Pardons, Natalie Kcomt Ché, Yves Vanrompay and Yolande Berbers
Katholieke Universiteit Leuven
Department of Computer Science
Celestijnenlaan 200A, 3001 Heverlee, Belgium

ABSTRACT

Home automation or domotics systems are ambient intelligence systems that are designed to help people proactively, but sensibly. In this paper we propose a system that learns and automates patterns in the interactions of the user with the home automation devices. We show our approach and architecture. An event processing tool is used to handle the events from the home automation devices, prediction algorithms predict the next action and both rule-based algorithms as reinforcement learning decide which actions are suitable to be automated. We show the results of our system on both a synthetic data set and a real data set. The automation system manages to automate a significant number of interactions for the user.

KEYWORDS

Domotics, ambient intelligence, embedded, user action automation

Copy Rights

1. INTRODUCTION

Today many homes are filled with home automation devices and sensors. These work together intelligently to increase the user's comfort by automating actions. The home is then called a smart environment. Mark Weiser described a smart environment as follows: "a physical world that is richly and invisibly interwoven with sensors, actuators, displays, and computational elements, embedded seamlessly in the everyday objects of our lives, and connected through a continuous network" (Weiser, 1991). Strongly related to smart environments is the concept of ubiquitous computing. Ubiquitous computing is a paradigm that represents computer technologies that are present everywhere and almost invisible to the user. A smart environment is an example of ubiquitous computing. To realize a smart environment the principles of ambient intelligence (AmI) need to be adopted. A possible definition for an AmI system is as follows: a digital environment that proactively, but sensibly, supports people in their daily life (Cook, 2009).

A person who lives in a house often uses the same devices every day and in the same order. The goal is to learn, recognize and automate these patterns of interactions with the various home automation devices. Automating user patterns and making predictions about future user actions increases user comfort and could make life easier for elderly and disabled. A classic example of such a pattern is to wake up, turn off the alarm clock and then make coffee. An intelligent system could learn this pattern and automatically make coffee in the morning, when the person turns off the alarm clock. The main assumption made by the system is that humans indeed follow such patterns which make reliable predictions possible.

To realize such an intelligent system the various domotics devices and sensors in the smart environment need to offer services over an intern network. Furthermore the inhabitants need to be observed for a while. Based on these observations the intelligent system can learn patterns, which can be used to predict and automate user actions. The main challenge for such a system is to predict and automate enough actions in a timely fashion, avoiding that the user manually has to start these actions, while preventing to automate actions that the user does not want. An intelligent learning system should be unobtrusive and when there is small noise, because of the users not following patterns, the system should not automate the action and should also detect and ignore this noise.

The following is a possible user scenario for an intelligent system. "Somewhere in a house in the future Jane wakes up and turns off the alarm clock. The curtains open automatically. Jane gets up and goes to the bathroom. The light turns on automatically as soon as she enters. The system has learned that she always showers in the morning and the water is heated to the preferred temperature. After showering Jane goes to the kitchen where her black coffee is already made. The system has learned that she prefers black coffee in the morning."

This paper is organized as follows. Related work is discussed in section 2. In section 3 we explain the architecture of the proposed system. Results of our experiments are presented in section 4. Conclusions and future work follow in section 5.

2. RELATED WORK

Several projects study the prediction and automatisation of user actions in a smart environment. There are three major projects that try to achieve this: The Adaptive House, iDorm and MavHome. The Adaptive House (Mozer, 1998) tries to automate lights, warm water, heating and ventilation. It uses artificial neural networks to predict the next state of the house. It balances energy cost and discomfort cost using Q-learning to determine which action to automate. The iDorm (Hagras, 2004) project tries to automate domotics devices in a dorm using techniques from fuzzy logic. Fuzzy membership functions are retrieved from the collected data. These are then used to construct fuzzy rules that capture the user behaviour. When a user overrides a setting that the system has automated, the weights of the fuzzy rules are adapted or new fuzzy rules are created. MavHome (Cook, 2003) uses Episode Discovery (ED) to discover frequent patterns in the usages of devices. Furthermore a prediction algorithm, Active Lezi (ALZ) is used to predict the next user action. The frequent patterns discovered through ED are used to improve the accuracy of the ALZ algorithm. The decision algorithm is TD(0) reinforcement learning, which determines what action has to be executed.

The architecture of the system presented in this paper is inspired on the architecture of MavHome. However, the system described here differs from MavHome in the use of an event processing tool to process events in a structured manner. Also other learning techniques are used. Here ED is not employed for scalability reasons and FxL and Jacobs-Blockeel are used as prediction algorithms instead of the ALZ algorithm. The decision algorithm also differs because in our work Q-learning, SARSA and rule-based algorithms are used instead of TD(0).

3. THE INTELLIGENT DOMOTICS SYSTEM

In this section the high-level architecture of the learning automation system (Kcomt Ché, 2010) is presented, followed by a detailed description of the various components.

3.1 High-level Architecture

The learning automation system needs to be able to send actions to the domotics devices. The principles of Service Oriented Architecture (SOA) are adopted for the creation of the services provided by the various domotics devices and sensors in the environment. To maximize platform independence these services are offered using WSDL, an XML based language to describe web services. Each of the devices and sensors need to offer its services using WSDL over an intern network. These web services make it possible to easily describe the services and allow devices of different manufacturers to communicate via this standardized language. It also works independently of the underlying hardware. Furthermore, the principles of Event Driven Architecture (EDA) are incorporated as they are used for systems that send events between independent software components and services. Events are generated by domotics devices and sensors in the environment and the learning system must be able to respond to these event. The communication in an EDA follows the publish-subscribe pattern, i.e. communication of events is asynchronously and possibly one-to-many.

Copy Rights

The architecture of the system presented in this paper thus uses a combination of EDA and SOA, called event driven service oriented architecture (ED-SOA) (Ghalsasi, 2009). Here EDA is used to expand SOA with the publish-subscribe communication pattern and the ability to collect events over a long period of time, after which they can be processed and correlated. SOA is needed because the domotics devices provide services to each other and to the user. EDA handles the events that the devices generate asynchronously and that need to be collected over a long period of time.

3.2 Detailed Design: the Information Layer

Figure 1 shows the design of the learning domotics system, consisting of four different layers. The physical layer contains the various sensors and home automation devices. The communication layer provides communication between the different devices. The information layer retrieves information and higher level abstraction from the sensors and home automation devices. Finally, the decision layer controls the devices. We will first discuss the information layer and then the decision layer.

The information layer includes the event processing tool, Esper (Esper), which receives, processes and forwards the events to the other components. Esper can do filtering, but also aggregates events to represent hierarchical or composite events. The prediction algorithms in this layer will predict the next action based on recent history. We use the Jacobs-Blockeel algorithm (Jacobs, 2003) and the FxL algorithm (Hartmann, 2007) for the prediction. Jacobs-Blockeel (JB) is an algorithm that is based on IPAM (Davison, 1998). IPAM uses a first order Markov model, i.e. the prediction made is based solely on the most recent event of the input sequence. Jacobs-Blockeel uses a mixed order Markov model instead to calculate the probability distribution for the next event. Initially first order Markov models are used. Whenever the algorithm makes a correct prediction, a higher order is activated, otherwise the order does not change. For example if the system correctly infers the rule that after observing 'a', 'b' will be observed then the system treats 'b after a' as a single observation. The advantage of using a mixed order Markov model is that high orders are only used when necessary, which benefits the storage and processing requirements. Jacobs and Blockeel claim that the highest order for the Markov model is not always the best choice to determine the probability for the next event. We chose for the Jacobs-Blockeel algorithm because it is based on IPAM, which yields very good results, but it also extends the algorithm by taking into account more than only the most recent event. In a home automation environment there are lots of possible actions to take and using a system such as IPAM that only takes the most recent action into account is not sufficient here. Furthermore the Jacobs-Blockeel algorithm mixes orders intelligently, treating frequently occurring sequences as single observations.

Copyright

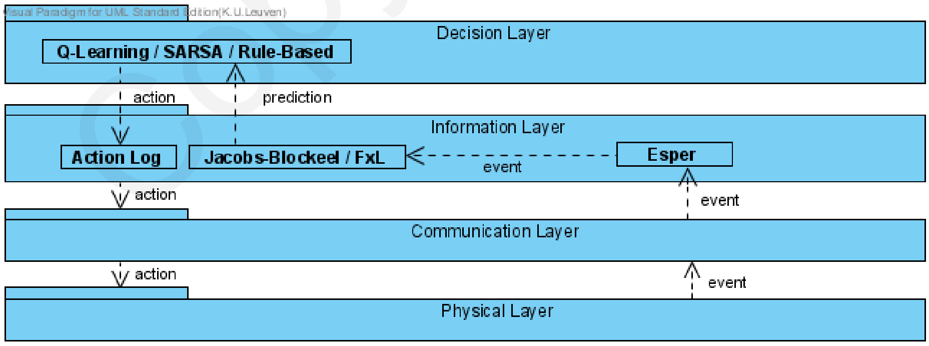


Figure 1. The architecture of the learning domotics system

FxL is an approach for combining the results of different order Markov models. The algorithm is based on an n-gram tree that contains the frequencies of different input subsequences with a length up to a specified value k. An n-gram tree is an ordered tree data structure of n-grams. An n-gram is a subsequence consisting of n items of a given sequence. These n-gram models are combined with weights to assign a score to each symbol, which represents the probability that a symbol appears next in the input sequence. AFxL is a variant of FxL that uses a slightly different weighting scheme. This relatively new algorithm yields very good results in the context of predicting commands in a Unix terminal. FxL was chosen because it is able to get these good results while keeping the storage costs limited by the specified k and the amount of possible user actions, whereas other algorithms like Active LeZi, with similar results, have storage costs that grow with the dataset size.

3.3 Detailed Design: the Decision Layer

The decision layer contains the decision algorithms to determine when to execute which action. To be able to keep the problem tractable, the decision algorithms can only decide to automate an action after an event has arrived from a domotics device or sensor. Two reinforcement learning algorithms were tested: Q-learning and SARSA. Also three rule-based algorithms were tested: based on a prediction-rule, distance-rule and confidence-rule.

The rule-based systems only use the prediction done by the prediction algorithm to determine which action to automate. Suppose *a* is the action that is predicted with the highest probability, P_a. The prediction-rule system will automate action *a* if P_a lies above a specified threshold parameter. Suppose that *b* is the action that is predicted with the second highest probability, P_b. The distance-rule system will automate action *a* if P_a - P_b lies above a specified threshold parameter. Suppose that the prediction algorithm made a prediction with a probability larger than a specified *eps* for *m* actions. The confidence-rule system (Vanrompay, 2010) will automate action *a* if $(P_a - 1/m)/(1 - 1/m)$ lies above a specified threshold parameter. A special case occurs when $m = 1$: then this formula is not used but simply P_a.

Next the reinforcement learning algorithms are discussed. The predicted action of the prediction algorithm is recorded in the state of the Q-learning algorithm together with the last events in the system and the time and date. The Q-learning algorithm is a reinforcement learning algorithm that is able to compare the different actions to take without explicitly modelling the whole environment with very good results. SARSA is a variant of Q-learning that does not make the assumption that the optimal policy can be followed. The reinforcement algorithm receives rewards by implicit or explicit feedback from the user. Based on these rewards the algorithm can learn to correctly automate user actions. Initially the algorithm takes random actions or the action that is predicted with the highest probability by the prediction algorithm to learn which actions in which states lead to good rewards. In a later stage the algorithm will prefer actions that lead to good rewards. The final action to be executed is sent by the decision algorithm to the home automation device in question. This layer works on top of the prediction system as an extra precaution to protect against possible errors the prediction system might make.

Copy Rights

4. EXPERIMENTAL RESULTS

The system was tested using different data sets. The first data set was created by using a synthetic data generator (Cook) and modified to support some extra options. The synthetic data generator generates events based on a user scenario. The scenario used here represents 70 days of a user living a repetitive life with no irregularities: the patterns during the weekdays differ from those during the weekend. The data set consists of 2582 events generated by 20 devices during this period. We will refer to this data set as SDG. The second data set is the MavLab data set. This data set is created by the MavHome (Cook, 2003) project and is based on real data from a real environment, namely a workspace consisting of different areas and equipped with various sensors and domotics devices. The data set was collected by gathering the events from the domotics devices and sensors during two months while six students worked in the lab on a regular basis. This dataset consists of 1362 events from 111 possible event types.

To evaluate the system we use two measures. The first measure represents the accuracy of the system. This is the ratio of the number of correctly automated events to the total number of automated events. The second measure is the correct automation rate. This is the ratio of the number of correctly automated automatable events to the total number of automatable events. Automatable events are events that represent interactions between the user and a domotics device that can be automated by the system. Both these measures are important and they are sometimes conflicting: a higher accuracy can lead to a lower correct automation rate.

Figure 2 compares the different prediction algorithms on MavLab while using the prediction-rule decision algorithm. The different accuracy and correct automation pairs are shown for the threshold parameter ranging uniformly from 0 to 1 with steps of 0.01. The data points are connected with a straight line as an interpolation to show where accuracy and correct automation rate would lie when using thresholds between the ones tested. Graphically the line of the best system lies to the right and above of the other lines. This isn't the case here: the lines cross each other. There is no clear winner. Jacobs-Blockeel is the best for higher accuracies while for lower accuracies Jacobs-Blockeel, FxL and AFxL lie closely together. IPAM performs significantly less than the other three systems, which justifies choosing (A)FxL and Jacobs-Blockeel. Jacobs-Blockeel and IPAM both clearly make a trade off between accuracy and correct automation rate when changing the threshold value. When the threshold parameter is high the accuracy is high, but the correct automation rate is low.

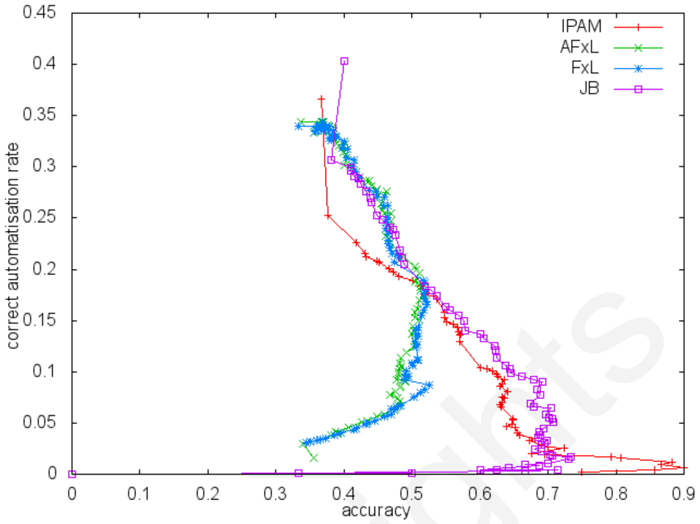


Figure 2. Comparison of prediction algorithms on MavLab using prediction-rule decision algorithm

Figure 3 compares the different decision algorithms on SDG while using the Jacobs-Blockeel prediction algorithm. Again the rule-based algorithms are shown by a line connecting the accuracy and correct automation pairs for various thresholds.

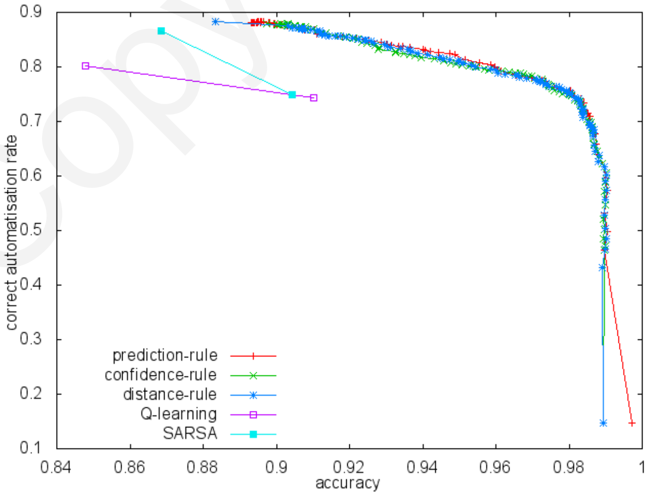


Figure 3. Comparison of decision algorithms on SDG using Jacobs-Blockeel prediction algorithm

The reinforcement learning algorithms do not make a clear trade off when changing a certain parameter. In our experience the results lie closely together. For Q-learning and SARSA we just show two data points: the one with the maximum accuracy and its correct automation rate and the one with the maximum correct automation rate and its accuracy. The results for Q-learning and SARSA are clearly worse than the three rule-based algorithms. Q-learning and SARSA itself lie closely together. The results for the three rule-based algorithms are quite high: it is for example possible to get 75% correct automation rate at 98% accuracy.

Copy Rights

These high results are explained by the fact that there is no noise in this data set. Between the three rule-based algorithms there is no clear winner: the lines lie closely together and intersect often.

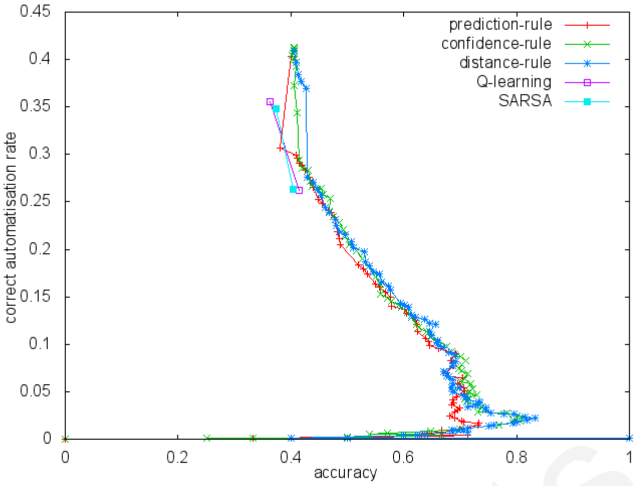


Figure 4. Comparison of decision algorithms on MavLab using Jacobs-Blockeel prediction algorithm

Figure 4 is similar to Figure 2 and compares the different decision algorithms on MavLab while using the Jacobs-Blockeel prediction algorithm. Again the reinforcement algorithms are worse than the three rule-based algorithms. There is again no clear winner between the three rule-based algorithms. MavLab is a real data set so the results are lower here: it is for example possible to get 10% correct automation rate at 66% accuracy.

5. CONCLUSIONS

We proposed an intelligent domotics system that learns patterns in the interactions of the user with the home automation devices and then automates these interactions. The architecture of the system is a combination of EDA and SOA, also called ED-SOA. For learning and automating user actions, we apply simple rules and various machine learning techniques: Jacobs-Blockeel, FxL, Q-learning and SARSA. These algorithms decide what action should be performed.

Our tests have shown that the AFxL and Jacobs-Blockeel prediction algorithms outperform IPAM. We have also shown that the reinforcement learning algorithms are outperformed by the rule-based algorithms. We suspect that this is mainly due to the large state space and too short exploration phase. We found no significant difference between the three tested rule-based algorithms. The combination of the Jacobs-Blockeel or AFxL prediction algorithm with a rule-based decision algorithm is extremely interesting for an embedded automation system. These two prediction algorithms were both designed to improve upon IPAM, while keeping memory requirements low and the rule-based algorithms keep no information in memory at all. Our tests have shown that it’s definitely possible to automate interactions with domotics devices in this way. However results on real data sets are not quite as good as on the synthetic data sets so further work is necessary to push these results higher.

In future work, the system will be tested live in a real smart environment. Furthermore additional rules should be tested and other information such as time and date can be taken into account by the rule-based algorithms.

Copy Rights

REFERENCES

Cook, D. et al, 2003. MavHome: An Agent-Based Smart Home. Proceedings of the First IEEE International Conference on Pervasive Computing and Communications (PerCom'03), pp. 521.

Cook, D. and Das, S., 2004. *Smart Environments: Technology, Protocols and Applications*. Wiley-InterScience.

Cook, D., 2009. Ambient intelligence: Technologies, applications, and opportunities. *Pervasive Mobile Computing*, Vol. 5, No. 4,pp 277-298.

Cook, D., *Synthetic Data Generator*. http://ailab.wsu.edu/mavhome/datasets/sdg.zip

Davison, B. and Hirsch, H., 1998. Predicting sequences of user actions. *Proceedings of AAAI-98/ICML-98 Workshop Predicting the Future*, pp. 5-12.

EsperTech Inc. *Esper, Complex Event Processing*. http://esper.codehaus.org/

Ghalsasi, S. Y., 2009. Critical success factors for event driven service oriented architecture. *Proceedings of the 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human*. Seoul, Korea, pp. 1441-1446

Hagras, H., 2004. Creating an Ambient-Intelligence Environment Using Embedded Agents. *IEEE Intelligent Systems*, Vol. 19, No. 6, pp. 12-20.

Hartmann, M. and Schreiber, D., 2007. Prediction Algorithms for User Actions. *Proceedings of ABIS* 2007, pp. 349-354.

Jacobs, N. and Blockeel, H., 2003. Sequence Prediction with mixed order Markov chains. *Proceedings of the Belgian/Dutch Conference on Artificial Intelligence.*

Kcomt Ché, N. et al, 2010. An intelligent domotics system to automate user actions, *Advances in Intelligent and Soft Computing*, Vol. 72, pp. 201-204.

Mozer, M., 1998. The neural network house: an environment that adapts to its inhabitants. *Proceedings of the AAAI Spring Symposium on Intelligent Environments*, pp. 110-114.

Vanrompay, Y. et al, 2010. An effective quality measure for prediction of context information. *Proceedings of the 7th IEEE Workshop on Context Modeling and Reasoning (CoMoRea) at the 8th IEEE Conference on Pervasive Computing and Communications (PerCom'10)*. Mannheim, Germany.

Weiser, M., 1991.The computer for the 21st century. *Scientific American*, Vol.265, No.3, pp. 66-75.

Copy Rights

A CROSS-FORMAT ARCHITECTURE FOR PROFESSIONAL PUBLISHING

Angelo Di Iorio, Antonio Feliziani, Luca Furini and Fabio Vitali
Department of Computer Science, University of Bologna

ABSTRACT

When generating content for high-quality output, the issue of converting data from sources of any quality comes across. Even with just text, the use of desktop tools to produce input for high quality processes such as professional publishing or e-learning material generation requires professionals to convert the result, adjust it and produce the final output object. Since conversion is usually unidirectional, re-use of content and minor amendments are prevented once the conversion is done, unless the same professionals are involved in all the production steps. This raises costs and stiffens the production process. In this paper we propose a flexible conversion architecture, an intermediate data format and automatic high-quality pagination tool that make the process of generating high-quality results from low quality sources almost automatic, and that require no quality adjustments after the conversion. The end result is that the original authoring tools can be used for all productions steps, thus returning the whole production process in the hand of the original author.

KEYWORDS

Digital publishing, multi-channel publishing, format conversion, ISA, IsaPress.

Copy Rights

1. INTRODUCTION

The production of *professionally formatted documents* is a complex task. In fact a heavy duty and cost intensive back office intermediates between the output of the author (most likely, MS Word files) and the input to the press (most likely, PDFs). Some of the activities of the editorial staff impact on the actual content of the book (such as proof reading, grammar check, etc.), some is legal/commercial, but many activities have much to do with the preparation of the final deliverable ready to go to the presses through pagination applications such as InDesign (InDesign 2010) or XPress (XPress 2010).

The output requirements in these situations are precise, high quality and imperative. This makes it impossible to use widespread word-processors to directly produce books ready for printing and distribution, although authors still prefer to write content with these tools, that are very simple, intuitive and not expensive (or even free). The presence of converters that produce PDFs from MS Word files does not solve this issue. Several of these tools are available, either commercial (CutePDF 2010) or free (Bullzip 2010). This list could go on and on. However, the output generated by such tools does not meet the requirements of a professional publishing house: the conversion models currently adopted are crude and approximate (Di Iorio et al. 2006).

The challenge is to provide a fully automatic conversion engine that generates high-quality results from arbitrary sources, removing the need for intermediaries to convert the input and revise the converted result.

In this paper we present a general framework that makes this process take place, focusing on the context of professional digital publishing. The overall framework relies on three main components: (i) a generic intermediate data format, called IML, (ii) a generic conversion architecture called ISA* and (iii) a sophisticated pagination engine. The paper is actually focused on the first two components with the intent of showing either their generality or their applicability to real scenarios. In fact, we go into details of IsaPress a cross-format architecture for professional publishing currently in use by an Italian Publishing House to produce high-quality books. Further details about the pagination engine are in (Di Iorio et al. 2006).

The paper is then structured as follows. Related works are investigated in section 2. Then we introduce our segmentation model and IML in section 4, and discuss ISA* in section 5. Section 6 is devoted to IsaPress, before conclusions in section 7.

2. RELATED WORKS

A basic principle is well accepted by the document engineering and markup languages community: the separation between content and format. This idea is so embedded and shared by the community that providing a complete list of references is impossible (canonical references are (Coombs et al. 1987) (SperbergMcQueen et al. 1997)). On the opposite side, we found very interesting ideas about the impossibility of separating content and presentation and actually segmenting documents reusable subcomponents. Hillesund (2002) argued that there is no way of separating content and presentation but they are strictly interconnected and mutually dependent. He consider the paradigm of XML "one input – many outputs" basically wrong and claims that is can be only substituted by a weaker "many inputs – many outputs". Indeed it is practically impossible to reuse content fragments and merge them from different sources into a good composite one. Walsh (2002) replied that position by stating that separation is possible either from a logical or practical perspective and by holding DocBook as example of the success of such distinction.

Unlike the global critique of Hillesund, Piez (2005) argued that in some context it could be useful and profitable to write documents taking in mind both content and presentation, and managing them as a whole unit instead of separated sub-components. According to the author, "there is no reason to fear or disdain presentation-oriented design, but designers need only to discriminate when they want and need an isolated layer for our information capture, and when they want to work more directly with the 'hot lead'".

Particularly interesting for an environment still heavily dependent on the circulation of physical copies is also this research (Norrie et al. 2005) about a content publishing framework for interactive paper documents, describing the use of special devices such as "digital pens" to interact with digital contents by means of printed materials. In other words, a particularly generated printed version of a document could be seen not just as a one-way output but as an input channel too, adding the advantages of digital information storage to the versatility of a paper sheet.

Copy Rights

3. DOCUMENT SEGMENTATION

To decouple the authoring process from the actual production of high-quality output, we propose a document segmentation model that expresses the most relevant constituents of a document, and upon which we have implemented advanced applications of document conversion.

3.1 The Pentaformat Model

Documents are traditionally segmented into *content* and *presentation* and, although opposite opinions exist (as discussed in the previous section), researchers and professionals agree on advantages of such approach. We refine this distinction by identifying five components that can be extracted from *any* document, regardless of its actual layout and presentation.

Table 1. The pentaformat model

Dimension	**Description**
Content	The plain information made of text and images (we mainly focus on these elements, and leave out audio and video for the moment).
Structure	The labels used to make the meaning and the logical organization of the content explicit. Both structure and content constitute the basic information written and organized by the author.
Presentation	The set of visual features added to maximize the impact of the document on human readers. Presentation is built over the structures and aims at strengthening what is inherently expressed by structured content.
Metadata	The set of information that make a document searchable, indexable and manageable within wider contexts.
Behavior	The set of dynamic actions triggered by events on a document.

Our model is called "Pentaformat" and summarized in Table 1. Our claim is not only that *any* document can be considered as the integration of those five dimensions, but also that they are clearly distinguishable from each other, and can be interchanged and reformulated to obtain different documents. In order to better explain the nature and impact of our segmentation model, some properties of these dimensions are discussed below:

- *Logical separation*: we consider each dimension as a partial perspective onto the same document. Each dimension provides specific information (orthogonal to all others), is created through the help of specific competences and has a specific role for the overall meaning of the document itself.
- *Mutual connection*: from a different point of view, these dimensions are strictly connected. They are built on the top of each other, and they "work together" for the overall meaning of the document.
- *Context-based relevance*: no hierarchy is imposed *a priori* over these dimensions, but they are equally important from a theoretical point of view, although the content can be probably granted some primality. It is the context that determines the relevance and replaceability of the other dimensions.
- *Context-based interchangeability*: depending on the context of use of the document, these components can be replaced with new ones. We can use the structure to fit the content into a completely different presentation, or express a set of metadata into a completely different vocabulary, etc.
- *Language independence*: information captured by each dimension can be expressed in different languages.

The actual instantiation of a dimension into a specific format does not influence the meaning of that information. Yet, the capabilities of a specific language limit what can be encoded with that language because of syntactical details independent from abstract specifications. In conclusion, a cross-dimensional property is necessary to complete our model: the *language* each dimension is expressed in.

So the real point of our work is to be able to separate and extract all constituents of a document so as to reformulate a few of them, or to reuse some of them in different contexts. To this end we look at producing *generic formats* that describe the specific constituent elements of each document, so as to facilitate understanding and reuse. A generic format is therefore a set of elements describing the relevant bits of the documents in terms of content, structure, presentation, behavior and metadata, although in this paper we will only refer to the first three dimensions.

3.2 IML: Intermediate Markup Language

From the multitude of languages, formats and documents we daily work with, we might conclude that a huge amount of complex and diversified structures are needed. In (Di Iorio et al. 2005) we proposed and discussed some patterns for descriptive documents, concluding that by adopting these and only these patterns, authors can write well-structured, complete and unambiguous documents easily. Table 2 summarizes these patterns.

Table 2. Patterns for expressing structures of digital documents

Pattern	**Description**
Markers	Empty elements, whose meaning is strictly dependent on their position
Atoms	Units of unstructured information
Blocks and inlines	Blocks of text mixed with unordered and repeatable inline elements that have the same content model
Records	Lists of optional, heterogeneous and non-repeatable elements
Containers	Sequences of heterogeneous, unordered, optional and repeatable elements
Tables	Sequences of homogeneous elements

That work helped us to demonstrate that a small set of structures is sophisticated enough to express most users' needs: it is always possible to write arbitrarily complex documents or to normalize existing ones into simplified versions by using only such a limited set of structural objects, yet still expressing the same information. These patterns can be adopted to design an abstract language to express the structured content of *any* segmented document. We created such language and called it IML (Intermediate Markup Language).

IML is syntactically a very simple language that maps these patterns into specific XHTML structures, and uses attributes to express extra properties. This approach is similar to the microformats (Khare 2006), which embed semantic information within texts, by using a set of simple and open tags and attributes. Unlike microformats, which use specialized tags for a specific context, IML is a general language that let users to express any kind of information by simply using few attributes. Instead of having a pre-defined and rich set of names which capture the meaning of a text, it proposes a flexible mechanism which can be used to model any content. At the end, few tags only compose IML: P (for blocks), SPAN (for in-lines), TABLE, UL, LI and DIV (for different containers) and few more, all characterized by the @class attribute.

Copy Rights

Since the objective of IML is expressing only *structured content* (i.e., expressing the role of each text fragment), with no presentational information and with no information about behaviour or metadata, IML documents are simply a sequence of content objects that simply specify which pattern each object respects (e.g., whether the object is a block text, a container, a table or an inline) and which specific class it belongs to (which kind of blocks it is, which level of nesting it has, and so on).

The innovation of IML does not rely on its tags and attributes, rather in the fact that a minimal and rigorous set of objects and rules actually made possible to implement automatic conversion and advanced publishing systems. Yet, some scenarios cannot be directly modeled with a so simple schema such as mathematical formulas, or graphical fragments, of forms, or fragments written in domain-specific syntaxes and so on. IML does not directly address them but can be easily extended for customized domains.

4. ISA*: A FLEXIBLE ARCHITECTURE BASED ON PENTAFORMAT

By combining IML and the segmentation model described so far, we have designed a simple architecture that can be (and actually has been) repeated for very different scenarios. We call it ISA*, since it generalizes some ideas developed for our previous project ISA (Vitali 2003). ISA is a web application designed to simplify and speed up the creation of web sites. Authors write content in MS Word (and specify the role of each text block by styles) and the system automatically converts such content into graphically advanced pages, by exploiting associations between the layout area names and the content styles, previously created by a graphic designer. ISA transforms such information into an XSLT that, in turn, will apply the selected formatting to the original content. ISA* applies a similar approach to heterogeneous domains and formats.

Basically, this architecture (shown in Fig. 1) separates all components of a document, then it works separately on each of them and then recombines them again for the final output. The whole system consists of bi-directional converters from and to any existing data format we want to support.

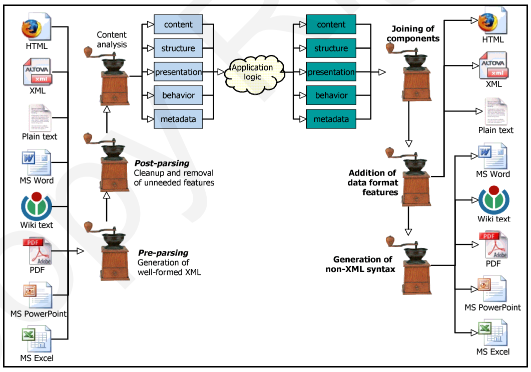


Figure 1. The ISA* architecture

Different models have been proposed for performing conversions between data formats (Mamrak & Barnes 1993): the *direct model* based on bidirectional transformations from a data format to another one, the *intermediate format model* based on a new format used as an intermediate representation of any format to be converted and finally the *ring model* in which data formats are virtually ordered in a circular structure and the transformation happens jumping from a format to the following one towards a pre-defined direction. The intermediate format model, a.k.a. *superior standard model*, has many benefits in terms of efficiency, quality and implementation facilities (Abiteboul et al. 1997) (Milo & Zohar 1998).

IML is therefore used as the intermediate data format that captures only relevant information of the input documents and ensures high-quality output by delegating the rendering to external powerful tools. Currently, the formats we manage include HTML, MS Word, ODF, PDF, LaTeX, plain text, as well as arbitrary XML. Note that IML is not meant to encode directly multimedia resources (such as EPS or JPG files, videos, etc.) that are referenced and attached to the document and need to be processed by external modules.

All document workflows using this approach follow the same general steps as shown in the picture: *content extraction* (on the left) and *high-quality post-production*.

Copy Rights

4.1 Content Extraction

The first step consists of extracting all the constituents of a document and normalizing content in IML. That segmentation process can be further divided in three steps:

4.1.1 Pre-parsing

Since our architecture is intrinsically based on XML, our fundamental tools are converters from one XML format to another. While some of these formats are already based on XML, others require a further step. Thus, the first action is converting the source format into XML, before any further content accommodation or interpretation takes place. This operation, called *pre-parsing* is heavily dependent on the actual syntax being employed by the data format being converted and it is heavily different among data formats.

4.1.2 Post-parsing

After the pre-parsing step we read the resulting XML file, clean up the content and remove the parts that surely will not be needed for the smart component separation. These operations, cumulatively known as *post-parsing*, are also different across data formats, but for semantic rather than syntactical reasons. These include re-joining lines into paragraphs, or removing alternative variants of the same image. Although these operations are still dependent on the quirks and peculiarities of each data format, they are done on an XML source, and therefore they can be and are usually done via a XSLT transformation.

Note also that adding a new format to the architecture requires implementing a new pre-parsing and post-parsing processor. The overall approach is still scalable as only these two modules depend on the input data format while the actual conversion logic is generalized.

4.1.3 Content Analysis

Pre-analyzed content is then scanned to identify individual features and denominate the constituents. The output of this phase is the same XML document that was provided in input, with additional attributes specifying whether an element is content, presentation, behavior or metadata. The current engine is based on XSLT technologies.

The key point is that any document can be passed to the engine, without imposing constraints on its internal structures and styles. The approach follows a *"Garbage In, Garbage Out"* paradigm: no input file is rejected, but the better structured is the source, the finer the final output will be. Note also that the analysis is fully automated and no user action is required.

We experimented different possibilities to ease and improve the quality of this information extraction. One extreme is to impose strict rules onto the authors, possibly enforced with macros that verify if they are following them. In this case, editing is not free of hassle, but the conversion is perfect, simple and straightforward. The opposite extreme is to have the system accept just any document and do its best to extract the actual content; in this case, the complete freedom in writing has heavy impact on the sophistication/complexity of the converted result. An intermediate solution, that we have tested in the e-learning context described in (Di Iorio et al. 2006b), consists in giving the users a set of guidelines about how to use styles and input macro, and then in implementing the appropriate transformations. Therefore, all documents can be processed by the system but the more compliant they are to the guidelines, the better will be the final result and the correct reformatting.

ISA* implements a hybrid approach. Whenever no guarantee of correct input is available, ISA* applies a set of heuristics and analysis techniques that allows users to provide even unformatted documents to produce well-formatted output (Vitali et al. 2004). These heuristics, expressed as parametric conversion rules, can be adapted so as to make this approach flexible for different scenarios and levels of complexity.

4.2 High-quality Post-Production

In the second step of ISA* architecture the perspective changes radically: what is an abstract description of content has to become an actual file, in a specific format, with specific formatting and layout. That process involves two clearly distinguished sub-processes: *application logic* and *high-quality rendering*.

Copy Rights

4.2.1 Application Logic

Once the five constituents of the document are separated, each specific application can act on them independently. Operations can vary considerably, from simple ones to rather complex ones, depending on the purpose of the application itself. Just a few examples are: (i) simply repackaging components in a different format in order to convert a document from one format to another, (ii) substituting the presentation constituent with a new one in order to reformat a document with a completely different layout or (iii) analyzing the structure constituent looking for specific types of content in order to filter it out. This list can obviously go on and on. It is worth noting, though, that all these operations are independent either of the input format or of the final one, since all files used by the application logic are all IML documents.

4.2.2 High-quality Rendering

Finally, all ISA* tools take care of re-generating a final document ready to be delivered to the final application. These processes follow a sequence absolutely symmetrical to the initial one: the new IML document is enriched with data format-specific information, and then converted via XSLT stylesheets into an XML format which can either be the final format, or the input to a converter to some kind of binary format, assuming we have the correct converter from XML to binary (such as a XSL-FO formatter).

Particularly important in this stage is the quality of the final conversion. The final rendering step takes in input both the converted document and configuration parameters that express the quality requirements to be met. By applying adaptive models, the renderer transforms the IML content into a ready-to-publish output, for instance a reusable and accessible learning object based on SCORM, or a sophisticated XSL-FO/PDF file.

Our model suggests then using external and format-dependent applications that are smart enough to take sophisticated decisions in order to generate high-quality results. The complexity inherent in such high-quality results depends also on the sophistication of the formatter that actually produces the final artifact. The more powerful and reliable is the renderer, the lesser is the effort required to produce high-quality products. However, in many cases the results that can be obtained with existing tools are not sufficiently sophisticated for professional use. For this reason, we often need to improve or re-implement renderers.

5. ISA* FOR PROFESSIONAL PUBLISHING: ISAPRESS

The abovementioned requirements are the milestones of IsaPress, an instantiation of the ISA* architecture for professional publishing. It is a system that automatically transforms unformatted content into ready-to-print and graphically advanced resources, in particular books. Assuring uniformity and high quality of their final products is not an easy and cost-effective task for publishing houses. Many manual interventions are still required to uniform source documents and make them ready to be "digested" by a (automatic or hybrid) conversion. The most widespread process to produce books involves different actors with different skills:

- *Authors*: they actually write content, ignoring the final formatting, and having few technical skills.
- *Publishers*: they decide the look&feel of the final product, in terms of formatting properties, dimensions, fonts, graphical choices and so on.
- *Pagination experts*: they transform the content provided by authors into a format ready to be processed and printed. They work with professional tools like InDesign (InDesign 2010) Quark XPress (XPress 2010) or PageMaker (PageMaker 2010) and perform some manual checks and corrections.
- *Typographers*: they actually print and bind the final books.

Note that we have omitted roles like proofreaders, reviewers, editors, etc. Our focus is not on the whole workflow of a publishing house, but rather on the semi-automatic conversion and publishing process.

In such a model, a leading role is still played by the pagination experts who are actually in charge of importing raw content in the system and verifying that they can be really transformed into a well-formatted book. According to the complexity of the final output, as well as the number of constraints to be fulfilled, such experts have even to manual intervene on the content and, when need, fix errors. Fact is, software formatters are still limited and do not solve automatically the most complex issues, publishing houses require complex properties to be satisfied, content is very often unforeseeable and full of exceptions, but without the work of pagination experts many books would not be published.

Copy Rights

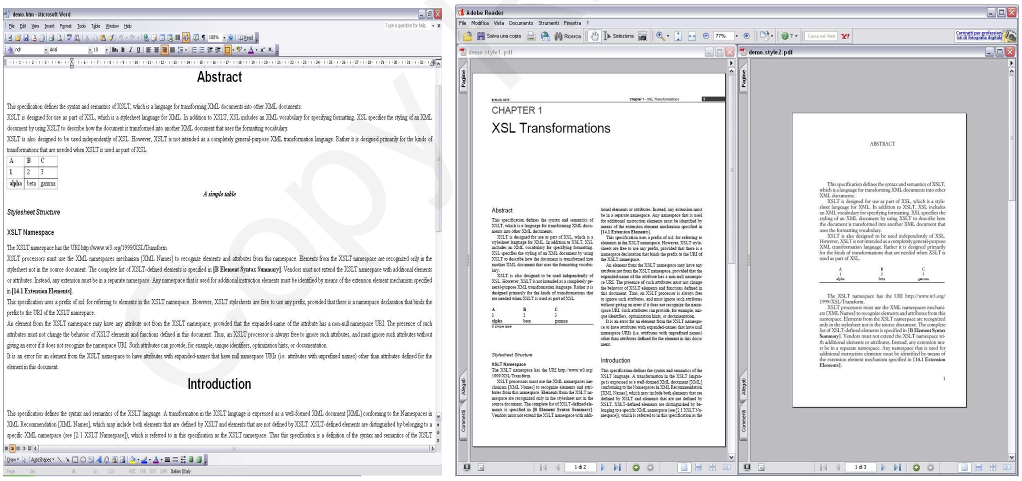
5.1 Revised Workflow with IsaPress

IsaPress addresses issues and limits of a traditional publishing workflow, by completely automating the production of high-quality books. The current implementation can be actually deployed in different ways: a stand-alone Java application, a web application and a module of a legacy CMS. The internal conversion engine relies on the principles discussed so far: allowing users to write content by using their personal productivity tools, segmenting the content into abstract components among which an IML representation of content, and running an automatic templating process that produces the final result.

Although IsaPress currently supports different data formats either in input or in output, the initial conversion goes from MS Word or ODF (OpenOffice) documents into PDF files. Since it has been the main focus of our research for a long period (and the ground where we obtained the best results), as well as one of the most useful applications, we use this conversion to explain concepts, achievements and possible extensions of our work. According to the ISA* architecture, the IsaPress process has two main steps:

1. *Content writing and extraction*: an author simply writes a document by using MS Word or OpenOffice. No particular plug-in is installed and no limitation is imposed over the tool features. The document is then processed by the IsaPress engine, which extracts its actual content and removes all presentational aspects, producing an IML file.
2. *High-quality post-production*: the intermediate IML document is then transformed into an XSL-FO file according to an XSL-FO template given in input. What is important here is the flexibility of the templating mechanism, which allows users to format the same content in very different ways without any further effort. It is enough to pass a different XSL-FO as input, and everything is automatically done by the system. At the end, the XSL-FO intermediate file is transformed into PDF, by exploiting the customized version of the FOP formatter described in (Di Iorio et al. 2006). The formatter in fact exploits some XSL-FO extensions and implements a revised Knuth algorithm (Knuth & Plass 1981) to automatically produce a ready-to-publish book.

Copyright


demo.htm - Microsoft Word
File Edit View Insert Format Tools Table Window Help
Type a question for help
100%
Read
ndr
Arial
10
Draw
AutoShapes
Page Sec At Ln Col REC TRK EXT OVR Italian (Italy
Adobe Reader
File Modifica Vista Documento Strumenti Finestra ?
Salva una copia
Ricerca
Seleziona
77%
Cerca nel Web
Contratti per professionisti di fotografia digitale
demo.style1.pdf
demo.style2.pdf
Pagine
Allegati
Commenti
1 di 2
1 di 3

Abstract

This specification defines the syntax and semantics of XSLT, which is a language for transforming XML documents into other XML documents.

XSLT is designed for use as part of XSL, which is a stylesheet language for XML. In addition to XSLT, XSL includes an XML vocabulary for specifying formatting. XSL specifies the styling of an XML document by using XSLT to describe how the document is transformed into another XML document that uses the formatting vocabulary.

XSLT is also designed to be used independently of XSL. However, XSLT is not intended as a completely general-purpose XML transformation language. Rather it is designed primarily for the kinds of transformations that are needed when XSLT is used as part of XSL.

A	B	C
1	2	3
alpha	beta	gamma

A simple table

Stylesheet Structure

XSLT Namespace

The XSLT namespace has the URI http://www.w3.org/1999/XSL/Transform.

XSLT processors must use the XML namespaces mechanism [XML Names] to recognize elements and attributes from this namespace. Elements from the XSLT namespace are recognized only in the stylesheet not in the source document. The complete list of XSLT-defined elements is specified in **[B Element Syntax Summary]**. Vendors must not extend the XSLT namespace with additional elements or attributes. Instead, any extension must be in a separate namespace. Any namespace that is used for additional instruction elements must be identified by means of the extension element mechanism specified in **[14.1 Extension Elements]**.

This specification uses a prefix of xsl: for referring to elements in the XSLT namespace. However, XSLT stylesheets are free to use any prefix, provided that there is a namespace declaration that binds the prefix to the URI of the XSLT namespace.

An element from the XSLT namespace may have any attribute not from the XSLT namespace, provided that the expanded-name of the attribute has a non-null namespace URI. The presence of such attributes must not change the behavior of XSLT elements and functions defined in this document. Thus, an XSLT processor is always free to ignore such attributes, and must ignore such attributes without giving an error if it does not recognize the namespace URI. Such attributes can provide, for example, unique identifiers, optimization hints, or documentation.

It is an error for an element from the XSLT namespace to have attributes with expanded-names that have null namespace URIs (i.e. attributes with unprefixed names) other than attributes defined for the element in this document.

Introduction

This specification defines the syntax and semantics of the XSLT language. A transformation in the XSLT language is expressed as a well-formed XML document [XML] conforming to the Namespaces in XML Recommendation [XML Names], which may include both elements that are defined by XSLT and elements that are not defined by XSLT. XSLT-defined elements are distinguished by belonging to a specific XML namespace (see [2.1 XSLT Namespace]), which is referred to in this specification as the XSLT namespace. Thus this specification is a definition of the syntax and semantics of the XSLT

© 88-08-8876 Chapter 1 - *XSL Transformations* 1

CHAPTER 1

XSL Transformations

Abstract

This specification defines the syntax and semantics of XSLT, which is a language for transforming XML documents into other XML documents.

XSLT is designed for use as part of XSL, which is a stylesheet language for XML. In addition to XSLT, XSL includes an XML vocabulary for specifying formatting. XSL specifies the styling of an XML document by using XSLT to describe how the document is transformed into another XML document that uses the formatting vocabulary.

XSLT is also designed to be used independently of XSL. However, XSLT is not intended as a completely general-purpose XML transformation language. Rather it is designed primarily for the kinds of transformations that are needed when XSLT is used as part of XSL.

A	B	C
1	2	3
alpha	beta	gamma

A simple table

Stylesheet Structure

XSLT Namespace

The XSLT namespace has the URI http://www.w3.org/1999/XSL/Transform.

XSLT processors must use the XML namespaces mechanism [XML Names] to recognize elements and attributes from this namespace. Elements from the XSLT namespace are recognized only in the stylesheet not in the source document. The complete list of XSLT-defined elements is specified in **[B Element Syntax Summary]**. Vendors must not extend the XSLT namespace with additional elements or attributes. Instead, any extension must be in a separate namespace. Any namespace that is used for additional instruction elements must be identified by means of the extension element mechanism specified in **[14.1 Extension Elements]**.

This specification uses a prefix of xsl: for referring to elements in the XSLT namespace. However, XSLT stylesheets are free to use any prefix, provided that there is a namespace declaration that binds the prefix to the URI of the XSLT namespace.

An element from the XSLT namespace may have any attribute not from the XSLT namespace, provided that the expanded-name of the attribute has a non-null namespace URI. The presence of such attributes must not change the behavior of XSLT elements and functions defined in this document. Thus, an XSLT processor is always free to ignore such attributes, and must ignore such attributes without giving an error if it does not recognize the namespace URI. Such attributes can provide, for example, unique identifiers, optimization hints, or documentation.

It is an error for an element from the XSLT namespace to have attributes with expanded-names that have null namespace URIs (i.e. attributes with unprefixed names) other than attributes defined for the element in this document.

Introduction

This specification defines the syntax and semantics of the XSLT language. A transformation in the XSLT language is expressed as a well-formed XML document [XML] conforming to the Namespaces in XML Recommendation [XML Names], which may include both elements that are defined by XSLT and elements that are not defined by XSLT. XSLT-defined elements are distinguished by belonging to a specific XML namespace (see [2.1 XSLT Namespace]), which is referred to in this specification as the

ABSTRACT

This specification defines the syntax and semantics of XSLT, which is a language for transforming XML documents into other XML documents.

XSLT is designed for use as part of XSL, which is a stylesheet language for XML. In addition to XSLT, XSL includes an XML vocabulary for specifying formatting. XSL specifies the styling of an XML document by using XSLT to describe how the document is transformed into another XML document that uses the formatting vocabulary.

XSLT is also designed to be used independently of XSL. However, XSLT is not intended as a completely general-purpose XML transformation language. Rather it is designed primarily for the kinds of transformations that are needed when XSLT is used as part of XSL.

A	B	C
1	2	3
alpha	beta	gamma

The XSLT namespace has the URI http://www.w3.org/1999/XSL/Transform.

XSLT processors must use the XML namespaces mechanism [XML Names] to recognize elements and attributes from this namespace. Elements from the XSLT namespace are recognized only in the stylesheet not in the source document. The complete list of XSLT-defined elements is specified in **[B Element Syntax Summary]**. Vendors must not extend the XSLT namespace with additional elements or attributes. Instead, any extension must be in a separate namespace. Any namespace that is used for additional instruction elements must be identified by means of the extension element mechanism specified in **[14.1 Extension Elements]**.

1

Figure 2. A source Word document and two very different PDFs from the same file

Figure 2 shows an example of conversion performed by IsaPress: a MS Word file has been transformed into two very different PDF files, both ready to be printed.

IsaPress is not a prototype, but a working system used by an important Italian academic publishing house, called “Il Mulino”, in order to officially publish books. More than one hundred books and journal issues have been published in the last three years, covering different subjects (economics, law, etc.) and including different objects: from statistical and tabular data to plain text, from pictures to complex tables, from hierarchical subsections to boxes and footnotes. The system has been also used to produce paper versions of e-learning material. Content is extracted from MS Word files and re-flowed into high-quality PDFs.

6. CONCLUSIONS

In this paper we have presented a general conversion architecture based on a simple, yet powerful, principle: segmenting documents into five components, working on them separately, and recombining them to obtain high quality output. In particular, content is normalized into an XML format, called IML. Besides the "many input -> IML -> many output" wide picture, we presented a specific end-to-end conversion, from Word files to high-quality PDFs, implemented by IsaPress.

Yet our main focus remains on supporting professional publishers. A full integration of the traditional workflow relying on well-known commercial tools is planned. Furthermore, support for a wider set of input and output option are foreseen: not just only MS Word and ODF, but also DocBook, XHTML, InDesign, Quark XPress and others to be identified.

REFERENCES

Abiteboul, S. Clouet, S, Milo T. 1997. "Correspondance and Translation for Heterogeneous Data", Proceedings of ICDT'97.

Barnard, D. T. and Ide, N. M. 1997. The text encoding initiative: flexible and extensible document encoding. J. Am. Soc. Inf. Sci. 48, 7 (Jul. 1997), 622-628.

Coombs, J.H, A.H. Renear, and S.J. DeRose, 1987. "Markup Systems and the Future of Scholarly Text Processing." Communications of the ACM, 30, 933-947.

Di Iorio, A., Feliziani, A. A., Mirri, S., Salomoni, P., & Vitali, F., 2006. Automatically Producing Accessible Learning Objects. Journal of Educational Technology & Society, 9 (4), 2006, 3-16.

Di Iorio A., Furini L., Vitali F., 2006 "A Total-Fit Page-Breaking Algorithm with User-Defined Adjustment Strategies". In the *Proceedings of the IS&T/SPIE Annual Symposyum on Electronic Imaging*, January, 2006, San Jose CA, USA.

Di Iorio A., Gubellini D., Vitali F., 2005. “Design Patterns for Document Substructures”. In the Proceedings of Extreme Markup Conference 2005, August, 2005, Montreal, Canada.

Hillesund, T., 2002. “Many Outputs Many Inputs: XML for Publishers and E-book Designers”. *Journal of Digital Information*, 3, 2002.

Khare, R. 2006. Microformats: The Next (Small) Thing on the Semantic Web? *IEEE Internet Computing 10*, 1, 68-75.

Knuth DE, Plass MF., 1981. Breaking Paragraphs into Lines. *Software. & Practice Experience*. 1981; 11(11).

Mamrak S.A., Barnes J., C. O'-Connell, 1993. Benefits of automating data translation, *IEEE Software*, July 1993, 82-88.

Milo T., Zohar S. 1998. Using Schema Matching to Simplify Heterogeneous Data Translation", *Proceedings of the 24th VLDB Conference*, New York 1998, 122-133.

Norrie, M.C., Palinginis, A. Signer, B., 2005. “Content Publishing Framework for Interactive Paper Documents”, DocEng’05, November 2–4, 2005, Bristol, United Kingdom.

Piez. W.“Format and Content: Should they be separated? Can they be?: With a counter-example”, 2005. In *Proceedings of the Extreme Markup Conference*, Montreal, Canada, 2005.

Sperberg-McQueen C.M. and Burnard L., 1997. A Gentle Introduction to SGML. In *Guidelines for Electronic Text Encoding and Interchange*, pages 13–36, 1997.

Vitali F., 2003. "Creating sophisticated web sites using well-known interfaces" in: *HCI International Conference*, Crete (Greece), 2003.

Vitali F., Di Iorio A., Ventura Campori E., 2004. "Rule-based Structural Analysis of Web Pages". In *Document Analysis System VI, Volume 3163 of Lecture Notes in Computer Science*, pp. 425-437, Springer Verlag, Berlin 2004.

Walsh, N., 2002: “One Input Many Outputs: a response to Hillesund”. *Journal of Digital Information*, 3, January 2002.

Tools and Web Resources

Adobe InDesign, http://www.adobe.com/products/indesign/, last visited 26 July 2010.

Adobe PageMaker, http://www.adobe.com/products/pagemaker/, last visited 26 July 2010.

Bullzip, http://www.bullzip.com/products/pdf/info.php, last visited 26 July 2010.

CutePDF, http://www.cutepdf.com/, last visited 26 July 2010.

FOP: Formatting Objects Processor. Available at http://xmlgraphics.apache.org/fop/, last visited 26 July 2010.

Quark XPress, http://www.quark.com/,last visited 26 July 2010.

Copy Rights

A SEMANTIC-BASED RECOMMENDER SYSTEM FOR THE SBTVD

Glauco da Silva* and Laércio Augusto Baldochi Júnior**
**Instituto de Aeronáutica e Espaço / Universidade Federal de Itajubá (UNIFEI) - Pça Mal. Eduardo Gomes, 50 - 12228-904 - São José dos Campos-SP – Brazil*
***Instituto de Ciências Exatas - Universidade Federal de Itajubá (UNIFEI) - Caixa Postal 50 - 37500-903 - Itajubá-MG – Brazil*

ABSTRACT

This paper presents an hybrid recommender system which exploits well succeeded strategies for recommending TV shows. Using reasoning techniques, we developed a content-based recommender that is able to discover knowledge about the user's preferences. We use this knowledge to build a profile that can be used (a) to recommend shows available in the TV schedule and (b) as an input to an external collaborative filtering recommender. When both recommenders work together, the provided results are significantly more effective, reducing overspecialization and producing unexpected but relevant recommendations.

KEYWORDS

Recommender systems, ontologies, spreading activation, interactive digital television

Copy Rights

1. INTRODUCTION

Information overload is the new buzzword of the 21st century. However, this is not a new issue: as far as 1994, in the beginning days of the World Wide Web, it was already a problem (Nelson, 1994). Of course, in our modern times, it becomes much more severe, as massmedia and global communication facilities become a pervasive presence in everyday life. As a result, people are striving to find the information, service or product they want, as it is more and more hard to distinguish what is relevant from what is irrelevant. As an attempt to address this problem, researchers developed the so-called recommender systems, computer programs that rely on information filtering techniques that attempt to present information items that are likely to be of interest to the user. Those information items may be news (Das et al., 2007), consumer products (Schafer et al., 1999; Huang et al., 2007) or TV shows (Ali and van Stam, 2004; Blanco-Fernández et al., 2008; Hölbling et al., 2010).

The Digital TV (DTV) domain has been gaining momentum, as it promises a new experience of watching television. In order not to make this new experience frustrating, the first step is providing ways of sifting through hundreds of channels to find interesting shows. Therefore, DTV has been considered an appealing domain for recommender systems. Dating back to 1999, TiVo (Ali and van Stam, 2004) was the first large scale commercial recommending system targeted to the TV domain. More recently, with the diffusion of Interactive Digital TV (IDTV) platforms, a myriad of new systems have been proposed. In general, these systems use well known recommendations strategies borrowed from the information retrieval literature, being categorized as content-based recommenders and collaborative-based recommenders (Adomavicius and Tuzhilin, 2005).

Content-based recommenders use features of a show, such as genre, cast or director as attributes for a learning system usually based in similarity metrics. As these metrics are based on syntactic approaches, it is only possible to detect resemblance between TV shows that share common attributes. This approach leads to overspecialization, suggesting shows that share attributes with other shows that the user have already liked. Therefore, overspecialization leads to expected recommendations.

Towards providing users with unexpected recommendations, collaborative-based recommenders use the strategy of recommending to each user shows that have been appealing to others with similar preferences.

Collaborative filtering techniques are effective to fight overspecialization. However, other difficulties arise from this approach. It is hard to make predictions for new users, since they do not have any personal data that can be used to calculate his/her neighborhood (the group of users with similar tastes). This is known as the *cold start* problem. Another problem arises when the user has very unusual and/or particular tastes – this user does not have any neighbors. This is known as the *grey sheep* problem. Finally, privacy concerns are also relevant, as the user's personal data needs to be collected.

Hybrid solutions, merging content-based and collaborative filtering techniques have been used to overcome the limitations of each approach alone (Burke, 2007). As content-based filtering solely exploits the features of the domain, it can be used to recommend items for new users and for people with unusual preferences. Still, these recommendations are poorer than those produced for "*regular*" users, as they suffer from overspecialization.

A more recently approach to fight overspecialization is based on reasoning techniques. Borrowing ideas from the Semantic Web, researchers are formalizing ontologies for specific domains, such as the IDTV, and applying content-based strategies that infers semantic associations among the user's preferences and the items available in the recommender system. Using reasoning techniques, this approach allows the discovering of knowledge about the user's preferences, which is then used to compare with items that can be suggested to the user. An important point about this strategy is that it allows the recommendation of shows that do not share common attributes with other shows that the user appreciated in the past. This is a clear indication that the resulting recommendations are not overspecialized.

In this paper we present an hybrid recommender system targeted to SBTVD, the Brazilian TVD System. SBTVD was built as a variant of the Japanese system, known as ISDB-T – Integrated Services Digital Broadcasting Terrestrial. In order to respond to local demands, the Brazilian government decided to develop a middleware for SBTVD from scratch. Named Ginga (Soares et al., 2010), this new middleware has opened opportunities for the development of new services and applications. Therefore, we are proposing a new approach for recommending TV shows. We foresee a recommendation process which is implemented in two distinct, but complementary phases.

The first phase runs in the user set-top box, on top of the middleware Ginga. Based on a semantic net created from a domain ontology built to represent the semantics of TV shows, our system apply spreading activation techniques in order to discover the semantic proximity from the TV shows that the user appreciated and others available in the TV schedule. As a result, the system creates the user profile, which can be used both to recommend shows to the user and to represent the users preference in the second phase of our recommendation process. It is worth noticing that phase 1 occurs solely inside the set-top box.

The semantic-based approach used to process content in phase 1 is effective towards not recommending overspecialized content. However, if the user expects to receive surprisingly recommendations, there is no scape from collaborative-based techniques. So, the second phase of our recommending approach uses an external server which receives user profiles and apply collaborative filtering techniques towards discovering, for each user, a set of shows that people like him/her enjoyed.

In this paper, we focus on the first phase of our recommending process. Section 2 presents our domain ontology, explaining how it is used to build a semantic net for each user. Section 3 explains the processing of the semantic net towards generating a user profile. Finally, Section 4 brings our conclusions and final remarks.

2. A DOMAIN ONTOLOGY FOR TV SHOWS

The multiplexing and service information aspects of the Brazilian Digital Terrestrial Television Standards are described in documents published by ABNT, the Brazilian Association of Technical Standards. One of those documents, the ABNT NBR 15603-2 (ABNT, 2007), defines the metadata that a transmitting channel is supposed to deliver together with the TV signal. Based on this metadata and influenced by the work of Naudet et al. (2008) and Blanco-Fernández et al. (2008), we manually designed an OWL ontology for TV programs. This ontology was created using Protégé tool (Noy et al., 2001).

The TV shows in our ontology are identified by specific instances belonging to a hierarchy of genres such as *Fiction*, *Sports*, *Leisure*, etc., which are organized in levels. For instance, *Sports* has two subclasses:

XGames and *Motor*. TV shows become instances of classes or subclasses of this hierarchy of genres, as it is depicted in Figure 1.

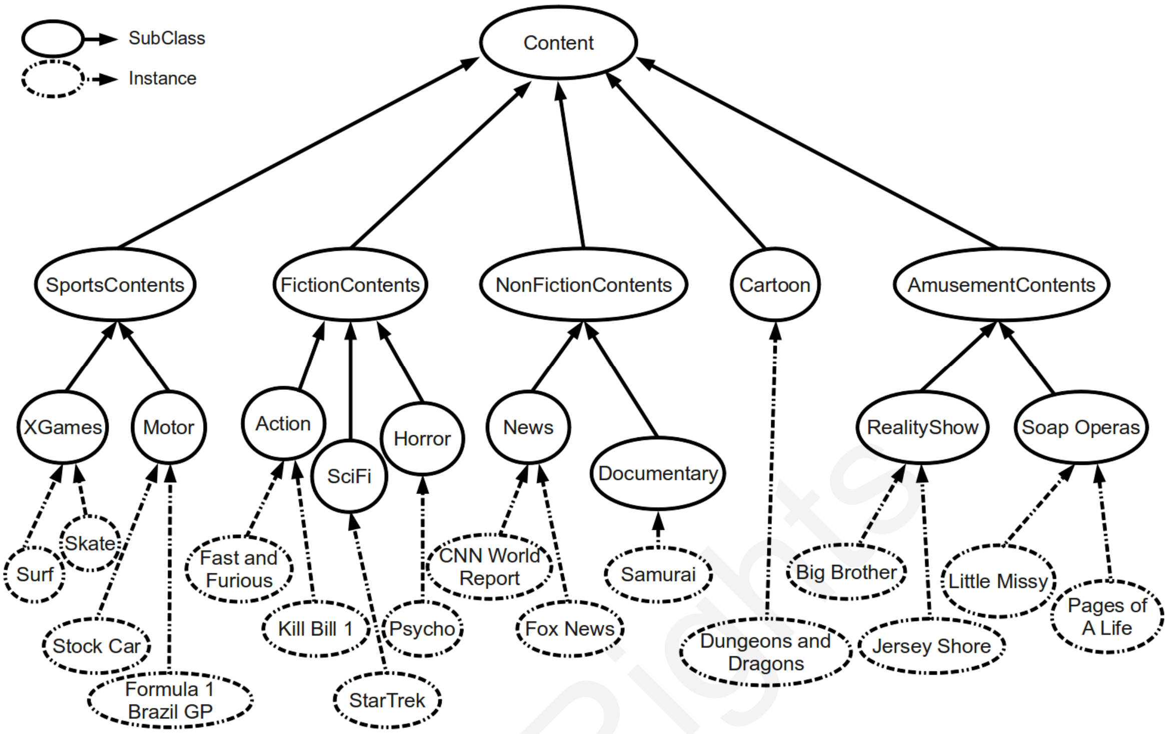


Figure 1. The TV ontology: subset of classes with instances

The instances of TV shows that appear on the bottom of Figure 1 are obtained by processing the TV schedule. The available programs are annotated with metadata extracted from Web sites such as IMDB – Internet Movie DataBase – and TheTVDB.com. For a given movie, it is possible to acquire information such as the genre, the cast of actors, the director, and so forth. All this collected information is used as semantic attributes for TV shows. They are represented in our ontology by classes which have their own hierarchies. What makes an instance of these classes a semantic attribute of another instance is a set of properties, such as *hasActor*, *isAbout*, *hasDirector*. Therefore, the semantic relationship among these instances produces a semantic net, as shown in Figure 2. According to this figure, *Inglorious Bastards* is an *Action* movie, about *war*. Therefore, it is semantic related to *Saving Private Ryan*, which is also an *Action* movie about *war*.

Our domain ontology was used as a basis for the construction of the semantic net. Thus, concepts and their relationships were obtained using classes, subclasses and relationships defined in the ontology. Following, we defined the weights of the relationships that link the concepts. Section 3.1 explains our approach towards defining these weights.

3. PROCESSING THE SEMANTIC NETWORK

As explained in Section 2, the semantic net is built using the TV Schedule and, therefore, contains instances of shows that the user has seen and (several) instances of shows that are available for consumption. Our goal is to recommend new shows among those available for consumption based on shows that the user has seen (and enjoyed) in the past. The first problem we had to address was determining the degree of interest of a user related to a show. In general, televiewers are not keen of rating shows explicitly. Therefore, we used an implicit way to measure this degree – I_i in (1) – in which we consider the relation among the time that the user has watched (consumed) a show (T_c) by the total time of the show (T_t).

Copy Rights

$$I_i = \frac{T_c}{T_t} \qquad (1)$$

In order to build the user’s profile, we arrange the TV shows in a priority queue, putting the most appealing programs to the user first, i.e., those programs that have a greater I_i. This approach allows a flexible control over the programs that are going to be used in the activation process.

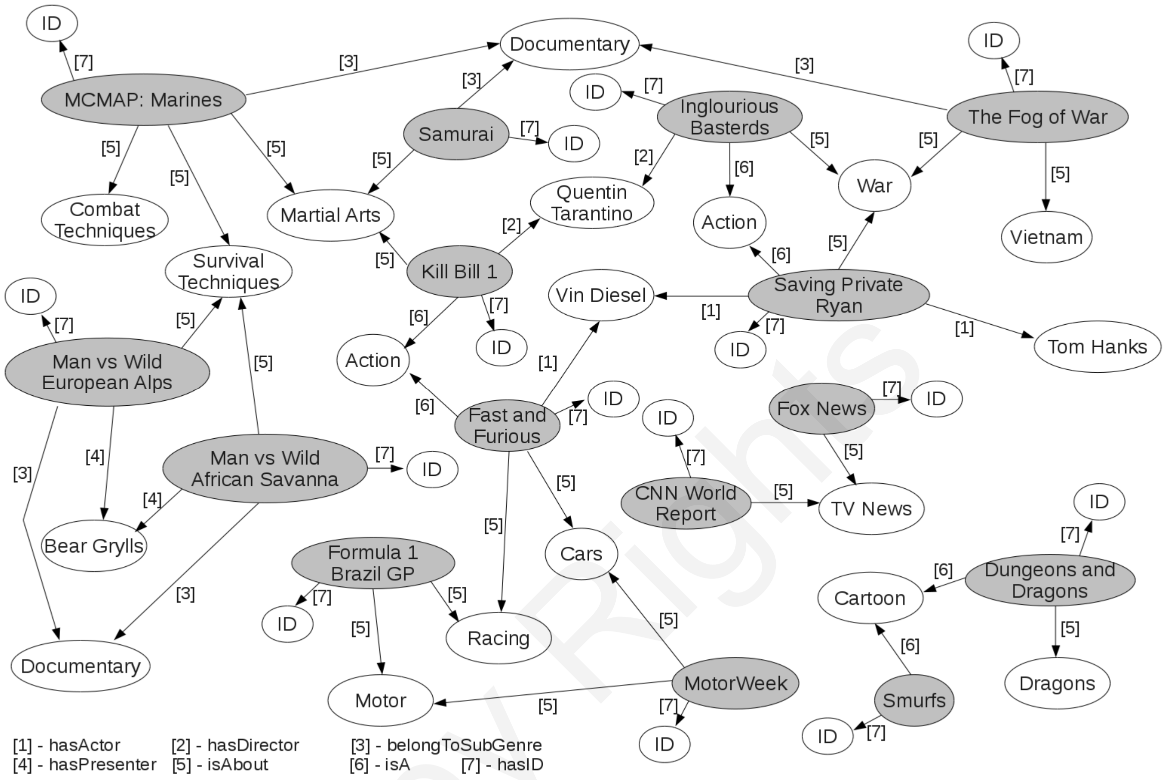


Figure 2. Fragment of a semantic network

Copy Rights

Towards understanding this process it is worth noticing the relationships that appear on the bottom of Figure 2. Each of these relationships has a weight empirically defined, which is higher or lower according to the semantic strength of the relationship. As an example, we will consider the relationships [2] *hasDirector* and [5] *isAbout*. Supposing we get the movie *Kill Bill 1* from our queue, with $I_i = 1$, as we traverse the semantic net, we find the relationship [2] connecting this movie to the director *Quentin Tarantino*. We also find another relationship [5] connecting this movie to the subject *Martial Arts*. Therefore, the initial activation is spread to neighbor nodes, causing the activation of these nodes. As a result, the documentary *Samurai* is also activated, as it is related to the subject *Martial Arts*. The same happens to the movie *Inglorious Bastards*. In this way, we are able to find semantic relations among items that do not present common attributes. In the next subsection, we present the internals of the spreading activation technique used in our system.

3.1 Semantic Spreading

Information about watched shows are kept in a log, which is processed from time to time (say, once a week). Processing this log, we get the priority queue, as we have explained in the beginning of this section. We can then determine a threshold to process this queue – the higher this threshold, the higher the affinity of the user to the consumed items.

The spreading activation will process items from the queue until the threshold is reached. Each item taken from the queue is used to activate the corresponding node in the semantic net, which receives an activation level equals to the degree of interest of the item. Following, this node's activation level is spread to its neighbors. The activation level of a reached node is computed by considering the levels of its neighbors and the weights of the links that join them to each other. Consequently, the activation level of a given node is directly related to the number of incoming links, the weight of those links and the relevance of the neighbors connected by those links. As an example of this, consider again the movies *Kill Bill 1* and *Inglorious Bastards*, depicted in Figure 2. Both of these movies are linked by the relationship [2] to the director *Quentin Tarantino*. Considering that a given user has watched and enjoyed these movies, their activation level (say, 1.0) would be spread to the director *Quentin Tarantino*, raising the activation level of this node. Therefore, the more quality links a node has, the higher will be its activation level.

Figure 3 depicts a situation in which several items (P_1, P_2, P_3, P_4, P_n) are semantically connected to a given item *I*. For this example, the activation value for *I* is calculated by formula 2

$$P_I(t+1) = P_I(t) + Q_j * w_{jI} * a \qquad (2)$$

where $P_I(t)$ is the activation level of item *I* in time *t*. Before the item's first activation, the value of $P_I(t)$ is zero. Q_j is the activation level of item *j*, the neighbor of the reached item, w_{jI} is the weight of the relationship that links *j* to *I*. Finally, *a* is an attenuation factor. Generalizing formula 2, we get the following formula

$$P_I = \sum_{j=0}^{n} P_I(j) + Q_j * w_{jI} * a \qquad (3)$$

where P_I is the activation level of item *I*, $P_I(j)$ is the current activation level of *I*, Q_j is the activation level of the neighbor of $P_I(j)$, w_{jI} is the weight of the relationship that links *j* and $P_I(j)$ and *n* is the number of items that links to *I*.

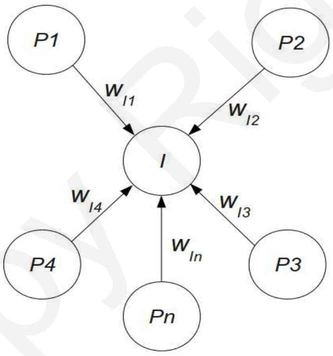


Figure 3. Computation of the activation level for a given node

The weights of the relationships were obtained empirically, based on results taken from the literature (Roth, 1999; Hussein and Neuhaus, 2010) and from a survey study conducted by Datafolha, a research institute affiliated to a major Brazilian newspaper. As this study (Datafolha, 2008) focuses on the consumption habits in the entertainment market, it was valuable towards finding the relevance of the attributes of a TV show, especially when movies are considered. Based on this information, we defined the relationships and their respective weights, as shown in Table 1. Using the results of the studies of Hussein and Neuhaus (2010), we also defined an attenuation factor of 5%. This value is used as an aging mechanism, attenuating the activation level of items at each new activation.

Table 1. Weights used in the ontology relationships

Relationship	Weight
hasDirector	0,30
isAbout	0,25
belongToGenre	0,20
hasPresenter	0,20
has Actor	0,15
actIn	0,15
belongToSubGenre	0,10

Copy Rights

3.2 Profile Generation

Our system produces a profile for each set-top box (STB). This decision takes into consideration the fact that TV systems do not have an implicit way of recognizing users, and forcing users to login and logoff is completely impracticable in a TV environment. Furthermore, as TV sets become increasingly cheaper, it is expected that in most of the cases, there will be a one to one relation among users and STBs.

As explained on Section 3.1, for each consumed content, we calculate a degree of interest, which is used in a spreading activation process. After processing all consumed items, the resulted semantic net presents several nodes containing activation levels greater than zero. These nodes represent information that is semantically related to the user consumption. Therefore, this is the information that will be used to recommend shows to the televiewer.

Instead of simply processing the semantic net towards finding relevant nodes that correspond to programs that are being aired, our approach extracts the information from the net, producing a profile. This profile can be used in an external collaborative filtering recommending system, which, after processing a group of profiles, sends back to the STB recommendations that are based on the consumption of similar users.

An important feature of our recommender system is that it can be used solely as a semantic content-based recommender system or as an hybrid system. In the first case, all processing occurs inside the STB, which brings advantages such as avoiding privacy concerns and data security issues. When those issues and concerns are not relevant, the second phase of our system is able to make recommendations based on the preferences of other individuals, which brings surprise to the recommendation process.

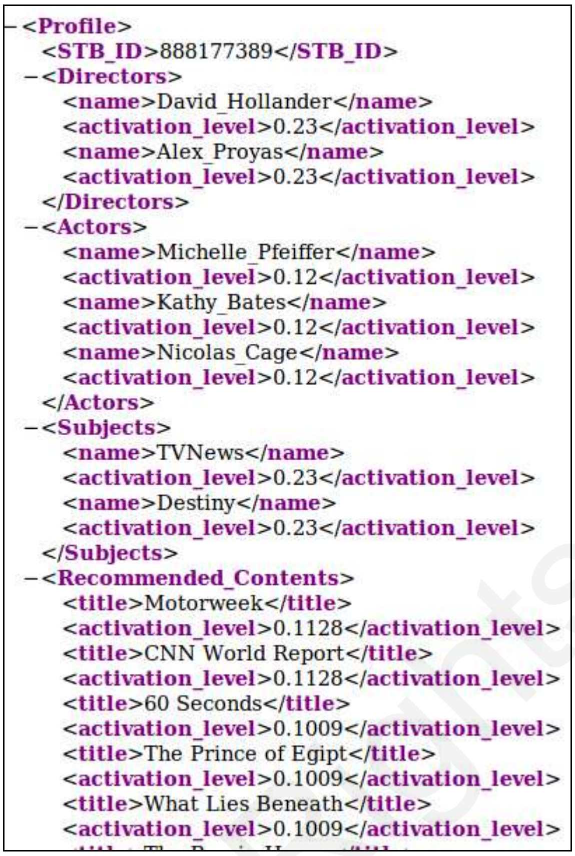
Figure 4 shows a portion of the generated profile for a given STB. A profile is an XML document containing the STB's ID together with information regarding the user's preference related to semantic concepts found in the TV domain, such as actors, directors, presenters, genres and TV shows, accompanied by their degree of interest.

When used as a semantic content-based recommender, our system process this XML profile, matching programs scheduled to be aired with those that appear in the profile with high activation levels (i.e., those that present a high degree of interest). The result can be presented to the user as a list containing the recommendations, grouped by the main genres of our ontology.

4. CONCLUSIONS AND FUTURE WORK

This paper has presented a recommender system which aims at integrating well succeeded strategies for recommending content. One of these strategies is the exploitation of reasoning mechanisms in order to overcome the limitations of traditional content-based recommenders. Another strategy is the use of collaborative filtering techniques, towards providing recommendations that are unexpectable but relevant at the same time.

Copy Rights



```
- <Profile>
   <STB_ID>888177389</STB_ID>
  - <Directors>
     <name>David_Hollander</name>
     <activation_level>0.23</activation_level>
     <name>Alex_Proyas</name>
     <activation_level>0.23</activation_level>
   </Directors>
  - <Actors>
     <name>Michelle_Pfeiffer</name>
     <activation_level>0.12</activation_level>
     <name>Kathy_Bates</name>
     <activation_level>0.12</activation_level>
     <name>Nicolas_Cage</name>
     <activation_level>0.12</activation_level>
   </Actors>
  - <Subjects>
     <name>TVNews</name>
     <activation_level>0.23</activation_level>
     <name>Destiny</name>
     <activation_level>0.23</activation_level>
   </Subjects>
  - <Recommended_Contents>
     <title>Motorweek</title>
     <activation_level>0.1128</activation_level>
     <title>CNN World Report</title>
     <activation_level>0.1128</activation_level>
     <title>60 Seconds</title>
     <activation_level>0.1009</activation_level>
     <title>The Prince of Egipt</title>
     <activation_level>0.1009</activation_level>
     <title>What Lies Beneath</title>
     <activation_level>0.1009</activation_level>
```

Figure 4. Fragment of the XML file representing the user's profile

As our system was targeted to run on a resource constrained set-top box, we decided to implement it in two distinct, but complementary phases. In the first phase, we developed a TV ontology which represents the semantics of TV shows. Next, we collected data from sites such as IMDB (www.imdb.com) towards providing semantic attributes for shows available in the TV schedule. Following, using the semantic relationships of our ontology, we build a semantic network. Finally, exploiting spreading activation techniques, we discover nodes in this network that are semantic related to items that the televiewer has seen and enjoyed. As a result, a profile representing the user is produced. At this point we can process this profile and recommend shows to the user, or we can send this profile to be processed by a remote program. That is when phase 2 starts.

In phase 2, we plan to implement a recommendation server that receives profiles from thousands of televiewers, cluster those users according to their preferences and recommend to each user TV shows that were appealing to others in his/her cluster. Developing a collaborative-filtering system is a challenging task, as there are complex issues involved. One of these issues is the problem of selecting the user's neighbors when the available preferences are sparse. This is a common problem for new users, who usually have small profiles. In our system, however, the activation spreading performed in phase 1 aids in the population of the profile. Even if the user has watched few shows, the semantic relationships among these shows and the concepts in our ontology helps the population of the profile, thus avoiding the generation of sparse profiles.

Copy Rights

REFERENCES

ABNT, 2007. *ABNT NBR 15603-2: Digital terrestrial television - Multiplexing and service information (SI) Part 2: Data structure and definitions of basic information of SI*. Associação Brasileira de Normas Técnicas.

Adomavicius, G. and Tuzhilin, A., 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *In IEEE Transactions on Knowledge and Data Engineering*, Vol. 17, No. 6, pp 734-749. doi: http://dx.doi.org/10.1109/TKDE.2005.99.

Ali, K. and van Stam, W., 2004. Tivo: making show recommendations using a distributed collaborative filtering architecture. *In KDD '04: Proceedings of the tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA, pp 394-401. doi: http://dx.doi.org/10.1145/1014052.1014097.

Blanco-Fernández, Y. et al., 2008. Providing entertainment by content-based filtering and semantic reasoning in intelligent recommender systems. *In IEEE Transactions on Consumer Electronics*, Vol. 54, No. 2, pp 727-735. doi: http://dx.doi.org/10.1109/TCE.2008.4560154.

Burke, R., 2007. Hybrid web recommender systems. *In The adaptive web: methods and strategies of web personalization*. Vol. 4321 of Lecture Notes in Computer Science, pp 377-408.

Das, A. S. et al., 2007. Google new personalization:scalable online collaborative filtering. *In WWW '07: Proceedings of the 16th international conference on World Wide Web*, pp. 271-280, New York, NY, USA. ACM, doi: http://doi.acm.org/10.1145/1242572.1242610.

Datafolha, 2008. *Hábitos de consumo no mercado de entretenimento (in portuguese)*. Available online at http://sedcmrj.locaweb.com.br/pesquisa/pesquisa_habitos_consumo_agosto2008.pdf.

Hölbling, G. et al., 2010. Personaltv. *Multimedia Tools and Applications*, Vol. 46, Nos. 2-3, pp. 259-288. doi: http://dx.doi.org/10.1007/s11042-009-0352-2.

Huang, Z. et al., 2007. A comparison of collaborative-filtering recommendation algorithms for e-commerce. *IEEE Intelligent Systems*, Vol. 22, No. 5, pp. 68-78. doi: http://dx.doi.org/10.1109/MIS.2007.4338497.

Hussein, T. and Neuhaus, S., 2010. Explanation of spreading activation based recommendations. *In Semantic Models for Adaptive Interactive Systems (SEMAIS)*, 1st Workshop in conjunction with the International Conference on Intelligent User Interfaces (IUI).

Naudet, Y. et al., 2008. An ontology-based profiling and recommending system for mobile tv. *In Proceedings of the Third International Workshop on Semantic Media Adaptation and Personalization*. Washington, DC, USA, pp 94-99. doi: http://dx.doi.org/10.1109/SMAP.2008.9.

Nelson, M. R., 1994. We have the information you want, but getting it will cost you!: held hostage by information overload. *Crossroads*, Vol. 1, No. 1, pp. 11-15. doi: http://doi.acm.org/10.1145/197177.197183.

Noy, N. F. et al., 2001. Creating semantic web contents with Protégé-2000. *IEEE Intelligent Systems*, Vol. 16, No. 2, pp. 60-71. doi: http://dx.doi.org/10.1109/5254.920601.

Roth, V. 1999. Content-based retrieval from digital video. *Image and Vision Computing*. Vol. 17, No. 7, pp 531-540. doi: http://dx.doi.org/10.1016/S0262-8856(98)00144-9.

Schafer, J. B. et al., 1999. Recommender systems in e-commerce. *In EC '99: Proceedings of the 1st ACM conference on Electronic commerce*, pages 158–166, New York, NY, USA. ACM.

Soares, L. et al., 2010. Towards the convergence of digital tv systems. *Journal of Internet Services and Applications*, Vol. 1, No. 3, pp 69-79. doi: http://dx.doi.org/10.1007/s13174-010-0002-y.

Copy Rights

SELF ORGANIZING MAPS FOR PUBLIC-CYCLING TRANSPORT MODELLING AND MANAGEMENT

Pablo Gay, Beatriz López, Albert Pla and Joaquim Meléndez
University of Girona
P-IV building
Escola Politècnica Superior
Universitat de Girona
Campus Montilivi
17003 Girona

ABSTRACT

Nowadays, many European cities are adopting public cycling solutions to improve citizen mobility. This kind of transport is based on the existence of bicycle depots where users take and return bicycles according to their necessities. The main challenge managing this type of systems is to guarantee the availability of bicycles and free dropping spaces, any time at any depot. This is faced by using additional depots as buffers and, at the same time, vans redistribute bikes among the empty depots to guarantee their availability. However, this solution creates the necessity to know and decide where to pick and where to unload the bicycles. In this work we propose to model the bike flow among depots using Self Organized Maps (SOM). SOMs are used to identify clusters classifying bike depots according to their behaviour so bicycle movements can be decided.

KEYWORDS

Modelling, management, transport, cycling, self-organizing maps, SOM, complex event processing.

Copy Right

1. INTRODUCTION

Public cycling is a promising alternative for citizen mobility across urban environments in towns and cities. The implementation of this kind of urban transport is fully aligned with the European Union's international commitments regarding the reduction of greenhouse gas emissions and European legislation on air quality (European Commission, 1999). In addition, bicycles offers an attractive individual transport alternative without constrain like fixed time-tables and predefined routes.

Some companies, aware of this new market arising from bicycle hiring, have started up to take advantage of this business opportunity. As a first step, they have defined several bicycle depots distributed in a given city. A citizen registered in as user (who is paying some rates for using the service) can take a bicycle from a depot A and return it to a different depot B in a maximum predefined period of time. This is for example the model used in Barcelona and Girona, both Spanish cities.

However, these companies have now some management problems related to the availability of bicycles and depots. Some users complain about the lack of bicycles at some depots because all the bikes have already been taken. On the other hand, sometimes users cannot place the bicycle in a depot because it is full. Consequently, companies are running fixed logistic rules to deal with those inconvenient and they introduce a complementary activity that consists in moving bikes from one depot to another every hour according to statistics and observations. But these kinds of policies are not enough. Crowd mobility and bike necessity is subject to many factors that are difficult to be considered together because of the inexistence of objective information related to them: working hours, weather conditions (sunny, rain), great events (soccer, pop concert, etc), proximity to train/bus/underground stations or business centres, etc. So, having an accurate model of bikes mobility is required in order to define the appropriate management policies.

Each town would have its own model, since every city has its own transport infrastructures that condition it. Thus, big cities with shops in the down town and industries at its surroundings would have different model from small tourist towns. Therefore, management rules should be defined accordingly.

Our research is concerned with both, modelling and management. In a first step, we aim to develop tools that support bike flow modelling. The results of this first step, feeds a decision making system for supporting management. As a first technique, we have explored self-organized maps to model bike mobility and k-nearest neighbour-like algorithms for decision making. The purpose of this paper is to explain our first prototype and results.

This paper is organized in five additional sections. Section two is devoted to the problem formulation and it is followed by the methodological approach used to solve it. Then, in section four experimental results using real data from Bicing, the cycling service of Barcelona (Spain) is described and discussed. Finally, we show related work and existing similar approaches in section five and main contributions are highlighted in the conclusion section numbered as six.

2. PROBLEM DESCRIPTION

Machine learning techniques offers a lot of possibilities to build models based on empirical data. In our case, we start with a database which contains the historic activity during a month in all the depots of the public-cycling transport in Barcelona known as Bicing (Bicing, 2010). The data base contains information about stations (identification, GPS coordinates and capacity information), users and dated hiring operations. Based on that information we formulate the problem as follows.

There are up to *n* bike stations (or depots). Each station is defined as follows:

Definition 1. A *station* is a tuple s_i = <gpsCoord$_i$, max$_i$, b$_i$ > where:

- *gpsCoord$_i$* are the coordinates (x_i, y_i), representing the cartographic latitude and longitude where the station is located.
- *max$_i$* is the capacity of the station, that is, the amount of bikes the station can store.
- *b$_i$* is the current quantity of bicycles at the station.

There are *m* users. Users are citizens registered in the bike hiring company. Each user has an identifier, id$_i$, and can perform the following operations: get a bike from the station (*checkOut*) and leave a bike at another station (*checkIn*).

Definition 2. A *checkOut(id_i,s_j,t)* is an operation performed by the user id_i at the station s_j at time *t*. The precondition of this operation is $b_j \neq 0$ and the consequence is $b_j = bj – 1$.

Definition 3. A *checkIn(id_i,s_j,t)* is an operation performed by the user id_i at the station s_j at time *t*. The precondition of this operation is $max_j – b_j \neq 0$ and the consequence is $b_j=b_j + 1$.

Thanks to this information, it is possible to infer user routes.

Definition 4. A *route* is a tuple r_x = <id_i,s_s,s_e,dur_x> where:

- id_i is the user identifier who has performed the journey
- s_s is the starting station
- s_e is the ending station
- dur_x is the duration spent on the trajectory.

Thus, a route can be computed as a consecutive *checkout – checkIn* operations performed by the same user. Formally:

Proposition 1. Given two consecutive operations, *checkOut(id_i,s_s,t_s)* and *checkIn(id_i,s_e,t_e)*, then, r_x=<id_i,s_s,s_e,dur_x> and ¬∃ *checkIn(id_i,s_k,t_k)* such that $t_s < t_k < t_e$ and ¬∃ *checkOut(id_i,s_k,t_k)* such that $t_s < t_k < t_e$.

From historical data, one can observe past failures in the bike management system when a station has either run out the number of bikes or it is full up.

Definition 5. A *runOut(s_i,t)* failure happens at a given timestamp *t* when there is station s_i has no available bikes for hiring. That is, ∃ s_i with $b_i = 0$.

Definition 6. A *fullUp(s_i,t)* failure happens at a given timestamp *t* when station s_i has no place for leaving bikes. That is, ∃ s_i with $(max_i - b_i) = 0$.

In this work we make the assumption that there is no central depot warehouse from which bikes can be moved to/from. This assumption fits the real case of the Bicing system in Barcelona in which we have

Copy Rights

centred the case study. Our goal is to develop the appropriate tools to avoid such failures based on the recognition of behaviours of users and developing policies to move bikes from one depot to another to avoid those possible blocking behaviours.

3. METHODOLOGY

In order to make decisions about how to move bikes, first we model the bike flow along a day. Then, we use the model to manage the bikes. This is then a two stage procedure.

3.1 Bike Flow Modelling with SOM

In order to model the bike flow without any further information that registers of the daily user operations, our first choice is to look for unsupervised machine learning methods. Particularly, we have considered using Self-Organized Maps (SOM) (Kohonen T., 1995) due they can reduce the complexity and the dimensionality of the inputs and helping us interpreting the results.

SOMs (or also called Kohonen map) are a competitive artificial neuron networks that has a single layers with the neurons distributed in a grid. In the learning phase, the units compete to learn the input data following the strategy (or variants) of winner takes all. It means the units specialize to represent specific input patterns and at the end of the learning procedure the neural network is capable represent the underlying model of the data presented. A visual example about how SOMs are structured can be seen at Figure 1. SOMs are an unsupervised algorithm, so while they are being trained, data vectors are introduced and compared with each neuron (represented by the input weight vector). Then, the neuron that has the minimum difference between input and weights (Best Matching Unit or BMU) and its neighbours are adapted (weight adjustment) to match better the last input.

In order to understand the mobility behaviour, we have reduced the scope of the study to a single day behaviour because rental periods defined in the system under study had a maximum duration of hours. Based on that assumption, we have defined the following methodology to identify sources and sinks of displacements at different times:

1. Split the historical data according to time intervals of one hour.
2. For every time interval:

2.1. Define a SOM with the following inputs: <station id, operation performed, timestamp>.

2.2. Colour the clusters obtained, and label them either with *sink* or *sources*. A *sink* cluster is one that mainly receives bikes at a given time interval. A *source* cluster is one where bikes are mainly dropped during a given time interval.

Note that a station can belong to both categories (sink cluster and a source cluster) depending on the time interval considered. But a given station can only belong to a single cluster at a given time interval.

Definition 7. $sourceC_i$ is a source cluster in the interval *i*. $sinkC_i$ is a sink cluster in the interval *i*.

Proposition 2. A station s_i in a given interval time *j*, either $s_i \in sourceC_j$ or $s_i \in sinkC_j$.

Note, then, that our station model is dynamic: for every time of the day a different map is selected. From the clusters, we can discover when a run out or full up will come from time intervals, and the clusters labels (source and sink correspondingly). Also, we can identify which stations have been run out or are in risk of being (as well as full up) and depending on the cluster types we can estimate the main routes the users use in the city, e.g. in the morning everybody goes to the centre to work, outskirts are source clusters an the downtown a sink. Thus in the decision making procedure, we can supply bikes from *sink* clusters, while filling bikes in stations of *sources* clusters.

Copy Rights

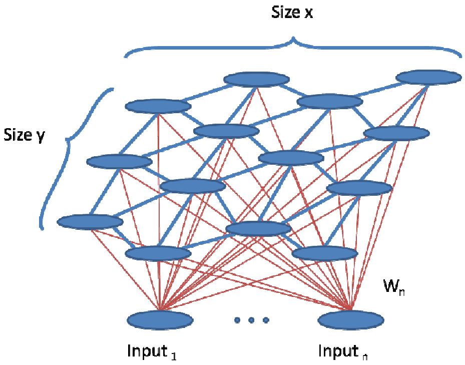


Figure 1. Example about how SOMs are usually structured. If we try to learn a model with *n* inputs, we need an *n*-length neuron array that acts as a competitive layer and is connected with all the other neurons in the classification layer.

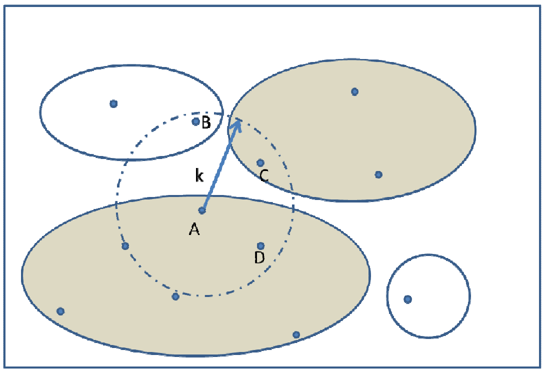


Figure 2. Concept of stations clusters at 10:00. Grey color indicates source clusters; white color indicates sink clusters.

3.2 Clustering-based Bike Management

Our ultimate goal is to avoid failures in the bike management. For that purpose we try to minimize the risk of failures by anticipating the supplying/removing bikes to/from stations. In case of failure, an alert message that helps to quickly solve the incidence is also helpful.

First, we define the risk of failure as follows:

Definition 8. *RunOut risk*: when there exists some station s_i with $b_i \leq \varepsilon$, where ε is a given parameter close to zero.

Definition 9. *FullUp risk*: when there exists some station s_i with max_i - b_i= ε, where ε is the same parameter than above.

Particularly, we denote by *runOutRisk(s$_i$)* and *fullUpRisk(s$_i$)* the risk on a station. It is a function defined in *{0, 1}*, such that:

$$runOutRisk(s_i) = \begin{cases} 1 & \text{if } b_i <= \epsilon \\ 0 & \text{otherwise} \end{cases}$$

$$fullUpRisk(s_i) = \begin{cases} 1 & \text{if } (max_i - b_i) <= \epsilon \\ 0 & \text{otherwise} \end{cases}$$

When either there is a risk or a failure, the supplying/emptying operations should be costless. That means that we should move bikes among depots close one to another. However, when moving bikes, special attention should be taken from which cluster we are moving from/to. That is, a solution to supply bikes to an empty station A could be the closest station B; however, if this station B belongs to a source cluster, it means that depot B is in a process of losing bikes. So we can solve the A problem while causing now a problem in B.

Taking advantage of the clusters obtained in the modelling step, we have defined a k-nearest neighbour procedure to decide bikes movements. Thus:

```
1.   Given a station $s_i$ with $runOutRisk(s_i) = 1$ at the current time $t$
2.   Select in $S$ the stations $s_j$ closer than a distance $k$ from $s_i$
3.   Determine the time interval $m$ corresponding to $t$ in the model.
4.   Remove from $S$ the stations belonging to source clusters
4.1. if $s_j \in sourceC_k$ then $S = S - \{s_j\}$
5.   Rank (in ascending order) the stations in $S$ according to their distance to $s_i$
6.   Choose the closest station $s_j$ to $s_i$
7.   Send a van to move bikes from $s_j$ to $s_i$
```

Figure 2 illustrates the method. In the figure, A is in risk of run out at 10:00 in the morning. When applying a distance *k* to know the closest stations, three stations are included in S: B, C and D. However, C and D belong to clusters that are also sources at that time; so they are removed from S. Thus, the station selected to supply bicycles to A is B.

A similar procedure has been defined for the risk of fulling up. However, in this case we take into account the station that has more room to allocate bikes. Thus the algorithm is as follows:

1. Given a station s_i with $fullUpRisk(s_i) = 1$ at the current time t
2. Select in S the stations s_j closed to a distance from k from s_i
3. Determine the time interval time k corresponding to t in the model.
4. Remove from S the stations belonging to sink clusters
5. if $s_j \in sinkC_k$ then $S = S - \{ s_j \}$
6. Rank (in ascending order) the stations in S according to their free places to s_i
7. Choose the station s_j at the top of the rank
8. Send a van to move bikes from s_i to s_j

In a future, we should also consider other optimizing approaches as a combination of distance and free places.

4. EXPERIMENTATION AND RESULTS

Due this project is indeed in an early development and experimentation stage, this first prototype of our methodology has been implemented mainly using the Java language programming. Although, we are using real previously stored, we are developing a solution to deal with that problem on line. Data has been provided by the company that is managing the cycling transport in Barcelona city (known as Bicing) and it contains the actual user routes. Therefore, for simulation purposes, we implemented a Complex Event Processing (CEP) system (Drools, 2010) for dealing with data. So this simulator generates events that are happening in the cycling system, i.e. the bike's check out and in described at section 2, which are gathered during time periods and then is used to train the SOM.

4.1 Experimental Set Up

The data used in the training stage of the SOMs contains the following information:

- Event type: Represents the event type. At this moment we are only looking after check in/out events.
- Station identification: That piece of information represents where the event happened.
- Timestamp: This is the temporal information that represents when the event happened.

But we found a problem while trying to work with all the data together because the information amount is too big (millions of events). Then, we decided to split it into intervals due, as we can see in Figure 3, it also can give us a good representation about event trends.

Specifically, we are experimenting with just one day information, split into smaller datasets according to a one hour interval, for example: events from 9am to 10am in one dataset, 10am to 11am into another and so on. We are developing studies with both working and fairy days and transitions between both, i.e. Friday and Saturday; although in this work we only reproduce results of one day.

Then the SOM structure we have created is a five sided square with hexagonal connection between neurons with a Gaussian neighbourhood function.

4.2 Results

Since we are still in an early experimentation stage, we have results mainly about the clustering which indeed shows that some patterns exist and can be modelled using a SOM. As example, Figure 3 shows us one of the clusters we have obtained from the data between 8am and 9am.

The Figure 3 represents a map of Barcelona with the depot stations from a sink cluster plotted as triangles. As we can see, almost all the stations are located in the city centre, specifically almost all the stations located to the west (left) are close to the Engineering University School and the Medicine University and the ones at the north-east (top-right) are in an office area plenty of big companies and close to the IL3 Master University.

Copy Rights

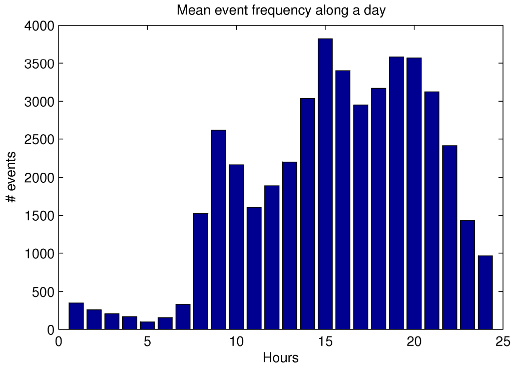


Figure 3. Event frequency along a day using bins one hour long. We can easily see how we have some very specific trends, e.g. at 9am, when people use to work/school/university, there is an event peak.

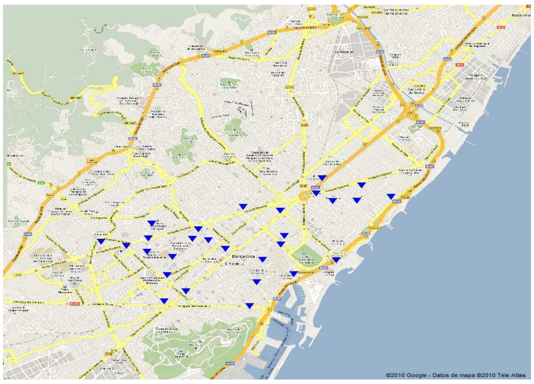


Figure 4. Map of Barcelona with a clusters from the 08:00 - 09:00 time interval. Triangles represent sink depot stations.

According to the bike management, as we said before, the project is in an early experimentation stage, so the tests we have run are still theoretical. What we propose is to introduce into the CEP system some rules that can be triggered when a station is nearly empty (or full). Then a call back to a previously trained SOM can be done in order to retrieve the proper station to interact with (refill or withdraw bikes) depending on the cluster the alarm was triggered.

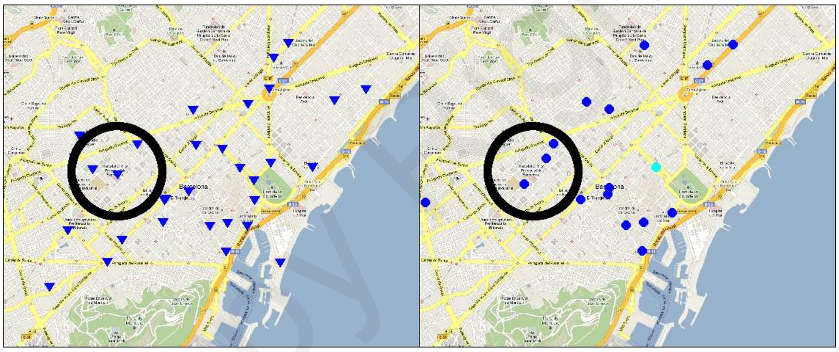


Figure 5. Example of the decision making management process. When a sink depot triggers the full up alarm a source station is fetched in a contiguous cluster within a k distance range.

Figure 5, shows what is still a work in process result. Once a run out alarm has been triggered by the system, for instance the sink depot inside the circle of the left image, the system fetches among all the other source clusters looking for the closest set of depots within a *k* distance. The image on the right shows a source cluster in a close time interval that could be used, so an optimal restore activity can be planned inside the supply van route.

5. RELATED WORK

Self-organized maps have been used in logistic is several work in order to analyse and segment the transport market. In this line, we would like to highlight (Kuo, R. et al, 2006) that combines SOM and k-means algorithms. In the first stage, the authors of (Kuo, R. et al, 2002) use SOM to obtain clusters that segment the market; however, in order to decide upon the appropriate number of clusters, then they interpret how many clusters can be useful, and in the second stage, they use k-means to obtain the final segmentation. The more recent work (Kuo, R. et al, 2006), the authors extend the k-means algorithm to a genetic k-means methodology in order to considerer different optimization criteria for determining clusters.

In (Tarca, I. C. et al, 2009) a comparison among fuzzy k-means, SOM, and the Gaussian method is performed in the problem of defining clusters that adequately distribute suppliers to costumers. From the results, SOM have been highlighted as a good tool to visualize density relationships in the logistics domain. A most promising work is (Hsieh, K.H. & Tien, F.C., 2004) in which the *location-allocation* problem is well defined. Such problem consists on determining appropriate supplier locations for optimizing the distances to the clients. In a future we need to have a look to this problem and try to fit how the location-allocation problem can be adapted regarding the dynamic behaviour of our cycling management problem. Event that in our case the locations are already know (we cannot change the coordinates of a bike station), we can improve our bike supplier decision with the insights performed in this field.

Regarding abnormal situations, it is important to distinguish the work of (Tardiff, S. R., 2004). The author proposes to extend SOM in order to include events, in what he calls self-organizing event maps. From that model, abnormal situations can be detected. In a future, we need to look to this work, and similar ones as (Hoglund, A. J. & Sorvari, A. S., 2000), to improve our model when several incidences happens at the same time.

Combining decision making and SOM is not new, although not so much used when symbolic information is treated. Thus, in (Hung, C., 2009), the authors shown how symbolic rules can be learn with SOM. Nevertheless, we are not using SOM for learning rules, but we are interpreting the outcome of SOM in an ad hoc decision making procedure for our specific domain. An interesting work that deals also with rules is (Bauer, D. et al, 2007). In this case, however, the author uses mathematical models to represent crowd flow in public transport. His main goal is to avoid collapses in subways and critical places. Fortunately, bikes do not suffer from those critical situations.

From the point of view of modelling, several previous works on crowd modelling are quite related to our research. For example, in (Sharma, S. B., 2007) the authors model the behaviour of students in a school; they use the concept of sink and source points as in this work, and deal with the resource usage and competition (stairs, elevators, etc.). The authors propose a two-stage simulator. In the first stage, they model the student's routes, while in the second one they introduce the spatio-temporal dimension. In this sense, this work is closer to us since we are also superposing different models to deal with the time dimension.

Another interesting work is (Andrade, E. L. et al, 2006b) where the authors propose hidden markov methods to model crowd flow; this models are then used to detect abnormal situations (see also (Andrade, E. L. et al, 2006a). Markov models are powerful tools that in the case of very well know scenes have a sense to apply. However, in the case of bicycling hiring, in which there are only a starting and ending point per user movement, flow is not so complex, and easier tools as SOM are enough. Most of the works, however, concentrate on congestions, as for example (Wohllaib, N., 2009).

6. CONCLUSION

This paper analyses one of the main problematic of the cycling public transport: the lack of bikes or the absence of free spots in the bicycle stations. The most common solution to this problem is to carry bicycles from one depot to another one with van or a truck but this solution presents the inconvenient of where to pick and unload the bikes and when the transfer must be done. To deal with this question the bike flow among stations is modelled with self-organized-maps.

This modelling offers a description about the behaviours of the different bike depots among time, showing in which periods a station is a bicycle source or a sink depot.

To avoid a failure in the system (when a depot runs out of bicycles or reaches its capacity), we propose a k-nearest neighbours procedure in order to show which bike movements should be done.

This methodology has been tested with data coming from a real bicycling company. The data corresponds to the Bicing activity in Barcelona during a full month. Since we are still in an early stage of experimentation, our results are focused on the clustering and bike flow modelling phase. As we expected, the SOM allow us to model the bike flow among time. Moreover, the resulting clusters can be associated with typical events of a city. E.g. we can observe how the number of check in events around a university grows during the class starting ours while the number of check out events increases when the classes finish.

In near future we plan to improve the decision making process taking into account other situations such as simultaneous risk warnings or multiple failures. Moreover we pretend to improve the modelling stage by

Copyrights

adding contextual information such as climatological data or special events information. We also consider the analysis of larger periods of time as users' behaviours is not the same during all the year.

ACKNOWLEDGEMENT

We would like to thank to Dídac Busquets the help on providing and understanding the experimental data. This research project has been partially funded by the Spanish Government and FEDER funds through the projects labelled TIN2008-04547, DPI2009-07891 and CTQ2008-06865-C02-02, DURSI AGAUR SGR grant 2009-00523 (AEDS) and BR10/18 Scholarship of the University of Girona granted to Pablo Gay.

REFERENCES

Andrade, E. L. et al (2006a), Modelling Crowd Scenes for Event Detection, in 'ICPR '06: Proceedings of the 18th International Conference on Pattern Recognition', IEEE Computer Society, Washington, DC, USA, pp. 175–178.

Andrade, E. L. et al (2006b), Hidden Markov Models for Optical Flow Analysis in Crowds, in 'ICPR '06: Proceedings of the 18th International Conference on Pattern Recognition', IEEE Computer Society, Washington, DC, USA, pp. 460–463.

Bauer, D. et al (2007), Macroscopic pedestrian flow simulation for designing crowd control measures in public transport after special events, in 'SCSC: Proceedings of the 2007 summer computer simulation conference', Society for Computer Simulation International, San Diego, CA, USA, pp. 1035–1042.

Bicing (2010), 'Bicing', http://www.bicing.cat/home/home.php.

Drools (2010), Drools 5 - The Business Logic integration Platform, http://www.jboss.org/drools.

European Commission (1999), 'Cycling: the way ahead for towns and cities', Luxembourg: Office for Official Publications of the European Communities, ISBN 92-828-5724-7.

Hoglund, A. J. & Sorvari, A. S. (2000), A Computer Host-Based User Anomaly Detection System Using the Self-Organizing Map, in 'IJCNN '00: Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks (IJCNN'00)-Volume 5', IEEE Computer Society, Washington, DC, USA, pp. 5411.

Hsieh, K.H. & Tien, F.C. (2004), 'Self-organizing feature maps for solving location–allocation problems with rectilinear distances', Computers & Operations Research 31, 1017–1031.

Hung, C. (2009), 'Knowledge-Based Rule Extraction from Self-Organizing Maps', Advances in Neuro-Information Processing , 139–146.

JBoss Community (2010), 'Drools', http://www.jboss.org/drools.

Kohonen, T. (1995). Self-Organizing Maps. Series in Information Sciences, Vol. 30. Springer, Heidelberg. Second ed. 1997.

Kuo, R. J. et al (2002), Integration of self-organizing feature map and K-means algorithm for market segmentation, Comput. Oper. Res. 29(11), 1475-1493.

Kuo, R. J. et al (2006), 'Integration of self-organizing feature maps neural network and genetic K-means algorithm for market segmentation', Expert Systems with Applications 30, 313–324.

Sharma, S. B. (2007), A static-dynamic network model for crowd flow simulation, in 'Proceedings, 6th International Space Syntax Symposium, Istanbul', pp. 101-01 – 101-04.

Tarca, I. C. et al (2009), Logistic system based on agents clustering using fuzzy and neural methods, in 'L. Sakalauskas, C. Skiadas and E. Skiadas and E. K. Zavadskas (Eds.): The XIII International Conference on Applied Stochastic Models and Data Analysis (ASMDA)', pp. 445-449.

Tardiff, S. R. (2004), 'Self-Organizing Event Maps', Master's thesis, Electrical Engineering and Computer Science, MASSACHUSETTS INSTITUTE OF TECHNOLOGY.

Wohllaib, N. (2009), 'Predictive Vision', Pictures of the Future. The Magazine for Research and Innovation. http://www.siemens.com/innovation/en/publikationen/pof_fall_2009/virt_real/crowdflow.htm (Accessed May 7th 2009)

Copyrights

EMERGENCY DETECTION BASED ON PROBABILISTIC MODELING IN AAL-ENVIRONMENTS

Bjoern-Helge Busch, Alexander Kujath and Ralph Welge
Leuphana University of Lueneburg, Institute VauST - Volgershall 1, 21339 Lueneburg

ABSTRACT

The actual demographic trend predicts a significant increasing percentage of elderly people in the German society until 2050. According to this trend, it must be assumed that the number of elderly persons solitarily living at home will rise as well. In this paper a human centered assistance system is tightly described and proposed as a resolve for the coherent challenges of this changes; the focus of consideration lies thereby on the statistical analysis of process data in order to derive an intelligent emergency detection based on probabilistic modeling. Using room automation events to design and train Hidden Markov Models for position tracking, complemented with the stochastically evaluation of telemedical devices and contactless sensor networks, the main issue is to achieve a solid, robust situation recognition mechanism. Due to the knowledge of the hidden user states or i.e. situations this approach offers the opportunity to detect emergency situations and to prevent harmful aftermath for the user through interventions like emergency calls. This paper illustrates the general functionality of the proposed solution and its key features by a vivid example. In addition to security aspects relating to the health status of the patient the recognition of the activities of daily life grants further advantageous options like energy management, comfort by assistance and building security completely adapted to people requirements.

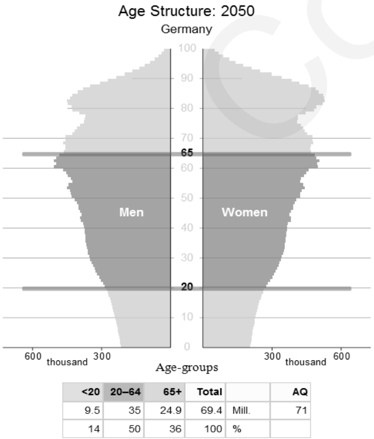
KEYWORDS

Probabilistic, situation recognition, Hidden Markov Models, distributed sensor networks, assistance systems

Copy Right

1. INTRODUCTION

Due to the demographic change the industrialized western countries are confronted with of an aging society. Nowadays in Germany 21% of the population are older than 65 years, but it is expected that this age group will grow up to 36% until 2050 (refer to figure 1). Some reasons for the alteration of the age pyramid are the aging of baby boomer generation, the overall increasing life expectancy, decreasing birth rates and an improved health care system. Due to this trend the growing number of elderly people living alone at home behaves diametric to the number of kin persons who cares for them. The demographic change will have a dramatic impact on the society, on financial aspects as well as on organizational aspects. The definition of *Ambient Assisted Living - AAL* by the BMBF/VDE comprehends technical concepts, products and services for the non stigmatizing, situation dependent support of people with special demands in their daily life under consideration of their informal self-determination. The integration of an AAL solution is provided, if constructive, completely technology-focused approaches fail due to their lack of consideration of human preferences and human environments. In order to enable elderly people to live in their familiar environment in a secured and comfortable manner, the institute *VauST* is working on user centered assistance systems, utilizing ambient technologies, as a part of the *AAL@Home* project.



<20	20–64	65+	Total		AQ
9.5	35	24.9	69.4	Mill.	71
14	50	36	100	%	

Figure 1. Age structure Germany 2050

2. CHALLENGES

2.1 Medical Challenges – Cardiac Diseases and their Impact for Society

Considering typical use case scenarios of the proposed assistance system in the field of geriatric care, the medical monitoring mechanisms take an important role for the patient's compliance, for postoperative therapy control or for the preventive detection of critical states of health. The decrease of the hospitalization rate is one goal of this approach. Due to their role as a cost driver in the health care sector (refer to figure 2), cardiovascular diseases like myocardial infarction, cerebral insults and heart arrhythmia or heart insufficiency are in the focus of further considerations. The prevalence and incidence of these diseases increased significantly in the last years and caused additional stresses and strains for the health insurance funds [Klauber et al, 2010]. Therefore it is one important requirement to implement reliable diagnosis functionality within the assistance system in order to reduce the costs by preventive detection of cardiac diseases. One essential problem in the context of many cardiac diseases like atrial fibrillation follows from the asymptomatic aetiopathology in many illness cases – the patient is apparently free of any disorders. Therefore a discreet continuous monitoring of vital signs like heart frequency for the identification of extrasystoles or tachycardia provides a great chance to apply necessary interventions to avoid serious aftermath before the patient recognizes any illness [Busch et al a), 2010].

rank	illness cases (men)	Number of patients	∅ hospital stay in days	∅ age
1	neonates by birthplace	245838	3.8	0
2	behavioral disorder by alcohol	233278	8.6	44
3	angina pectoris	177595	5.2	65
4	heart insufficiency	156893	11.5	73
5	Hernia inguinalis	148363	3.7	56
6	chronic ischemic heart disease	144579	6.1	66
7	myocardial infarction	134721	8.8	66
8	virulent neoplasm of bronchia	131461	8.2	66
9	Intracranial harm	123417	4.3	33
10	Pneumonia	112508	9.9	56
11	Atrial fibrillation	107623	5.6	65
12	Cerebral insult	101254	12.9	70

Figure 2. The most common hospitalization causes in Germany

2.2 Technical Challenges – Limitation of Actual Telemedical Devices

Human centered assistance systems, which implement a continuous vital sign monitoring, are subjects of numerous restrictions:

- Users may reject or have reservation towards the system, because camera-based systems imply an impression of observation to the user.
- Restrictions of user's mobility and autonomy, if body attached sensor-networks are applied.
- Concerns towards data privacy, if external service providers are connected to the system.

Studies with modern telemedical devices are proving an increased detection rate of cardiovascular diseases [Hördt et al, 2003], but they reveal drawbacks in the line with domestic care by outpatient care, because they fail due to the limitations mentioned before. Therefore telemedical ECG recordings are restricted to the checkup of arrhythmia and applied to technologies like cardiac pacemakers, ICDs and event-recorder [Koudi et al, 2006]. For human centered assistance systems it is mandatory to use contactless measurement methods for the data acquisition, if there is a claim to represent an appropriate alternative to existing systems.

Copy Rights

3. METHODS AND TECHNIQUES

3.1 Major Tasks of Assistance Systems

Human centered assistance systems are subsumed together with other technology-concepts, products or services which serve a situation-dependent support of people with special needs. Our concept of an assistance system regards some essential tasks considering the integration of the user: *Perception-, Access-, Communication-* and *Cooperation-assistance* [Busch et al b), 2010]. The assistance system supports the user in the accomplishment of their daily needs through decentralized technical services and their adaption to these needs. The detection of individual user situations, including their preferences also, is essential for the ubiquitous providing of appropriate, cooperative services for comfort, safety, energy awareness and security.

3.2 Architecture of the Assistance System

The kernel of the architecture for the proposed assistance system relies to the hierarchical arrangement of techniques based on probabilistic functions and description logic. The topology of the design is generic and adaptive to different use cases (refer to figure 3). The *process layer* comprehends all accessible information about the observed area of the specific use case. The process data covers all information about the user's environment which can be obtained e.g. by distributed embedded sensor networks (for detail refer to section 3.3). The *data acquisition layer* contains proceedings and filter operations, which are needed to gain and transform the essential measurement signal. The *metadata layer* handles the mapping of the acquired process data and the unspecified part of the collateral information to a generic data structure. The *data fusion layer* offers a conglomerate of methods and techniques to derive higher knowledge for a superior grade of integration from the basic information extracted in the lower layers or to increase the quality of these values. The estimation of situations, esp. of the internal hidden states of the system, depends on uncertain process data and incomplete knowledge. In this layer the extracted emission data and the a priori knowledge form the basement for the modeling of the discrete event systems (DES) according to the spatial and temporal environments of our assistance system using the advantages of HMMs. The *probabilistic modeling* for this specific scenario is discussed is in the following sections. The layer of the *semantic integration* deals with the transfer of the explicit and implicit knowledge about the structure and the identified states of the addressed domain into a representation of description logic. The implemented reasoning service gathers higher knowledge for the situation recognition attending to the associated knowledge base by cyclic queries and using appropriate description logical interferential mechanisms. Application-specific rules follow within the corresponding applications, which are running within the context of the assistance system and are assigned to the *application layer*. The *situation recognition* accesses the reasoning service in a cyclic query to analyze the different events and states due to the probabilistic modeling. The retrieved conclusions about the inner states are used for the completion of the application rules, e.g. emergency assistants, which decide about appropriate actions relating to the recognized situations [Busch et al b), 2010].

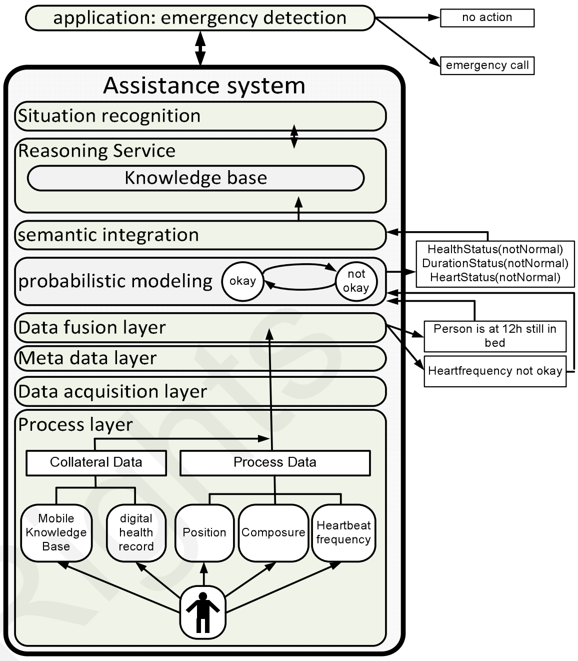


Figure 3. Architecture of the assistance system

3.3 Process Data Acquisition

There are three different kinds of technical devices which are utilized for the acquisition of the process $\{y_n(t)\}, n \in N, t \in R_+$ values relating to the inner states of the observed system. Considering the sampling time Ts and discretization effects like distortion by noise the process data can be expressed by $\{y_d(nT_s)\}$ with $d, n \in N, T_s \in R_+$. This data results from these three essential procedures:

- Evaluation of telemedical devices such as sphygmomanometer, mobile ECG recordings etc.
- Evaluation of the home automation components
- Evaluation of contactless sensor networks particularly UWB-radar

The telemedical equipment will be used by the patients themselves after stipulated intervals which relate to the anamnesis stored in the digital health record (DHR). Partially this information is gathered at deterministic points of time nT_s via standardized communication interfaces like Bluetooth, stored and expanded through a specific data structure and conveyed to the data fusion layer as described in section 3.2.

Copy Rights

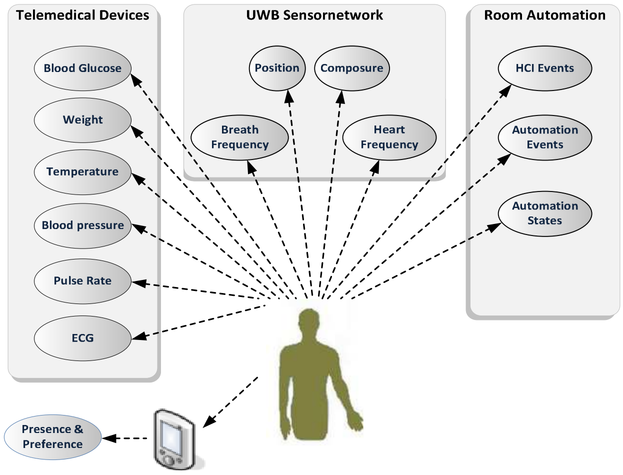


Figure 4. Types of process data

Another source of system information refers to the examination of the home automation components including features like system states or events just such as the opening/closing of a door for example. The preferred home automation system is accessible through the established interface BACnet and is evaluated in a cyclic query. Contactless measurement takes an exposed tank in the context of assistance systems implementing medical monitoring functionality as described in section 2.2. One promising concept is based on the work of [Helbig et al, 2007] and deals with the detection of heart rate and breath frequency through UWB-sensor components. Another approach proposes the appraisal of composure and position with the aid of UWB-sensor networks [Thomä et al, 2007]. In summary, the relevant process values gained in the data acquisition layer can be expressed by these following terms:

$$\{y_{assistancesystem}\} = \{\{y_{telemedicalDevices}\}, \{y_{UWB}\}, \{y_{Automation}\}\} \text{ with } \{y_{Automation}\}\} = \{e_{HCI}, e_{BASC}, x_{BASC}\}\}$$
$$\{y_{UWB}\} = \{f_{Heart}, f_{Breath}, (x,y,z)_{Position}, \ldots\} \text{ and } \{y_{Telemed}\} = \{\Theta_{Body}, m_{Body}, (mmHg)_{systolicBloodpressure}, \ldots\}.$$

These properties are individually evaluated due to specified parameters which relate to system inherent characteristics including the patient's characteristic traits (refer to figure 4).

3.4 Probabilistic Modeling with Hidden Markov Models

Hidden Markov Models (HMM) combined with Relational State Descriptions are a successful approach to design software agents addressed to temporal and spatial environments [Meyer-Delius et al, 2007]. An alternative approach deals with the sensor based activity recognition utilizing Relational Markov Network (RMN) under inclusion of the MCMC-algorithm for inference [Liao et al, 2005]. The works of [Taskar et al, 2002 and Pearl, 1997] deal with the implementation of RMNs with the aid of undirected graphic models i.e. Markov Nets. Furthermore the hierarchical expansion of dynamic Bayesian Networks is a probate resolve for probabilistic modeling [Subramanya et al, 2006]. The work of [Rabiner, 1989] reinforced the theoretical aspects of HMMs and their benefit for technical solutions in the context of statistical modeling. A Hidden Markov Model, representing a stochastic process like a time corresponding motion pattern, may be described by the quintuple $\lambda = \{X,A,Y,B,\pi\}$ with the state space X, the alphabet Y, the relating characteristic emission and transition matrices B and A, and finally the initialization vector π. The behavior of the probabilistic model refers to the relationship expressed by the equation $P(X_{t+1}=x_j|X_t=x_i)$, which is drawn for every state transition by the matrix A. As sketched in figure 5 every hidden state emits an element of the subset Y due to distribution specified by the matrix B. For further detail refer to the following sections.

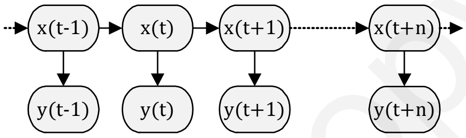


Figure 5. State/emission sequence

4. IMPLEMENTATION EXAMPLE

4.1 Subgraph for Position Recognition

The assistance system evaluates distributed sensor networks which are embedded within the home of the patient. One problem to solve is the recognition of the place where the patient dwells. The position is one indicator for the recognition of the person's activity and essential for the estimation of the user situation.

Copy Rights

Therefore one subgraph was drawn to reflect the position tracking through the habitat by the evaluation of the home automation events. The positions are considered by the state space:

$$X_{Home} = \{x_1,\ldots,x_n\} = \{bedroom, study, floor, storage\ room, bathroom, living\ room, kitchen\}$$

with $\{1,2,3,\ldots|X_{Home}|\rightarrow X_{Home}$. Every room is equipped with automation devices such as motion sensors m_x, door contacts d_x and window contact w_x (refer figure 6). The evaluation of these devices implies an alphabet $Y_{Room} = \{w_{11}\ldots w_{72}, m_3, d_3, d_{76}, \varepsilon\}$ under consideration of situations without notice of any emission through the surrogating symbol ε. If a person uses a door or opens a window, a significant emission is registered and used to approximate the most likely location of the person. The relationship between the stochastic process model and inhabitation is illustrated by the Markov graph (refer to figure 7). Relating to the motion track drawn in figure 8 we get a state sequence

$$Q_\tau=\{1,1,1,1,1,1,1,5,5,5,5,6,6,7,4,7,7,7,6,3,3,3,3,3\}$$

for an evaluation interval $T_{eval} = nT_s$ with $0 \leq n \leq 24$. We obtain a corresponding emission sequence formally expressed by $O_T = \{o_1, o_2,\ldots, o_T\}$, $o_t \in Y_{Home}$ in order to determine the hidden states:

$$O_T = \{\varepsilon, \varepsilon, \varepsilon, \varepsilon, w_{11}, w_{12}, d_{51}, \varepsilon, \varepsilon, w_{52}, d_{56} \ldots\}$$

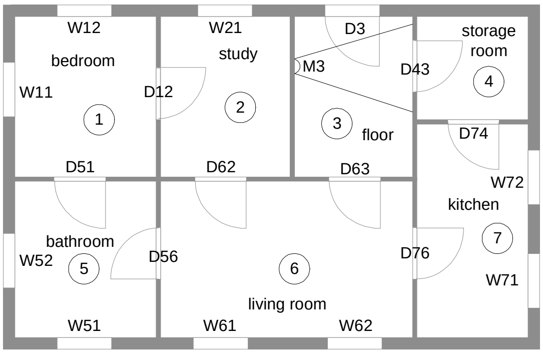


Figure 6. Footprint of the mock habitat

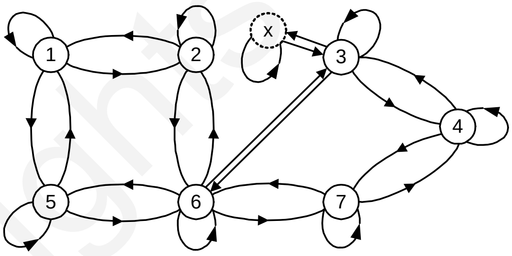


Figure 7. Markov graph representation

The relationship between emission sequence and the state sequence follows from the equation:

$$P(O_{To} \mid Q_{To}, \lambda) = \sum_{for\ all\ Q} [X_t, X_{t+1},\ldots, X_{To} = i, O_T\ldots O_{To} \mid \lambda]$$

The Viterbi-algorithm expressed by the term

$$\delta t(i) = \max_{q_1,q_2,\ldots,q_{t+1}} P[X_t, X_{t+1},\ldots, X_{To} = i, O_t\ldots O_{To} \mid \lambda]$$

offers the opportunity to approximate the most likely state of the system. With an explicit state assignment for the model with $\pi_{eval} = \{1, 0, 0, 0, 0, 0, 0)$ and previous initializations $\delta_1(i) = \pi_i b_i(k = o_i),1 \leq I \leq |X|$, $\psi_1(i) = 0$, we use the recursive equations

$$\delta t(j) = \max_{1\leq i\leq|X|}[\delta_{-1}(i)a_{ij}]b_j(o_t)$$

$$\psi t(j) = \arg\max_{1\leq i\leq|X|}[\delta_{-1}(i)a_{ij}]$$

$$X_t^* = \psi_{t-1}(X_{t+1}^*), t = T-1, T-2,\ldots,1$$

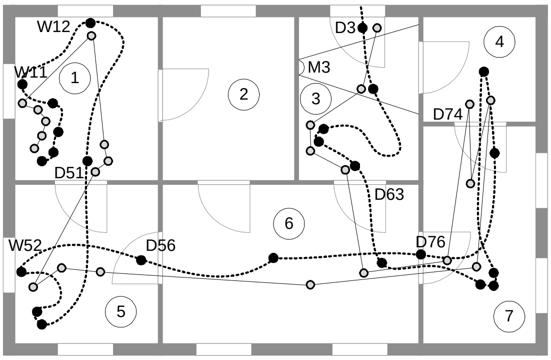


Figure 8. Example track

Through backtracking it is possible to identify the most likely path through the trellis structure. Referring to figure 8 the black dots mark the real positions of the example motion sequence, the grey dots mark the estimated positions. Untrained models, recognizing only the geometry of the mock apartment, permit a rate of cognition of 36% only. Models, trained with real recorded emissions and plotted state sequences with approved techniques like Baum-Welch, offer an optimized average detection rate of approximately 85%.

4.2 Activity Recognition for Feature Extraction Adjustment

The knowledge of the most likely activity of the patient is essential to choose the best fitting parameter set to obtain the emissions for the probabilistic modeling. The estimated position is one criterion for calling the adaptive activity HMM-graph. For example, if the person resides in the bedroom, a graph with a state space

Copy Rights

$X_{bedroom}$ = *{Sleeping, Dressing, …}* is selected. Or if the person’s location changes to the bathroom, a model with the state space $X_{bathroom}$ = *{showering, usingToilette, bathing, shaving, …}* is active and responsible for choosing the correct parameter structure (refer to figure 9). The training and evaluation of these stochastic activity models is done in the same manner as the position tracking. The process data, gathered and filtered in the data acquisition layer is used to gain significant emissions (refer 3.2-3.3). For example, the repeated measurement of the body temperature is obviously used for the detection of the indicator feaver:

$$\Theta_{Body}(nTs) > \Theta_{body}(Param) \rightarrow_{Ot=nTs} = Feaver$$

In analogy the detection of overweight works alike:

$$\hat{m}_{Body}(nT_s) > \hat{m}_{Body}(Param) \rightarrow O_{t\,=\,nTs} = Overweight$$

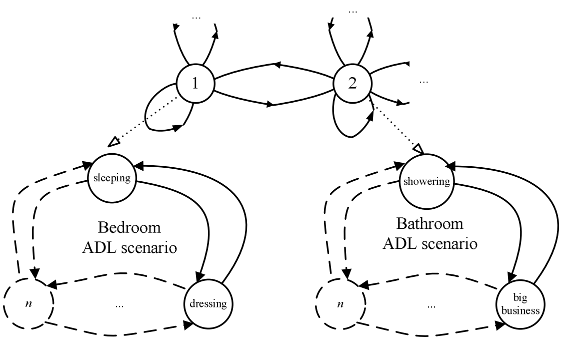


Figure 9. Relation between location and activity model

The boundary values for each type of vital sign classification depend on user’s activity at the one hand and on the other hand on information stored in the parameter set. These parameter sets are usually predicated on the anamnesis, the age, gender, fitness and other collateral information stored within the digital health record. According to the estimated activity the interpretation of the vital signs is essential for the recognition of critical situations. Complemented with the criterion of state duration it is a useful approach for emergency detection and prevention of greater harm. For example, if the typical user with an age of 65 dwells in the bedroom at 2:00 AM, it must be assumed, that this person is sleeping. The mechanisms of the assistance system choose the left partial model in figure 9 and select the appropriate parameter set (refer to figure 10). This implies that the average breath frequency and average heart rate should be lower than by day when the person is active. In this example, the permanently monitoring of the heart rate through an embedded UWB-radar component, which is installed in the bed, offers a reliable method for the detection of critical heart events like arrhythmia or other indicators for serious situations. The recognition of emergency cases is the main issue of this assistance system.

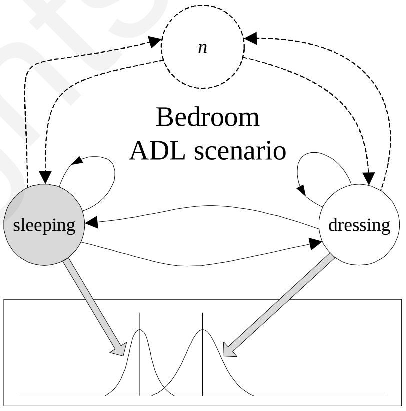


Figure 10. Boundary value selection

Copyrights

4.3 Emergency Detection

The ability to recognize dangerous situations enables preventive interventions like emergency calls to the hospital, notification of the attending doctor, messages to the members of the patient’s family or to take appropriate actions like the dispense of anticoagulants or other advisable medicine in danger. One descriptive example deals with the preventive detection of atrial fibrillation. This cardiac disease is one predisposing factor for cerebral insult and therefore indirectly in charge for a large amount of health costs in Germany (refer to section 2.1). Atrial fibrillation may be drawn by the HMM

$$\lambda_{AF} = \{X_{AF}, A_{AF}, Y_{AF}, B_{AF}, \pi_{AF}\}$$

with the state space

$$X_{AF} = \{No\text{-}AF,\ 1st\text{-}AF,\ Paroxysmal\text{-}AF,\ Persistent\text{-}AF,\ Permanent\text{-}AF\}.$$

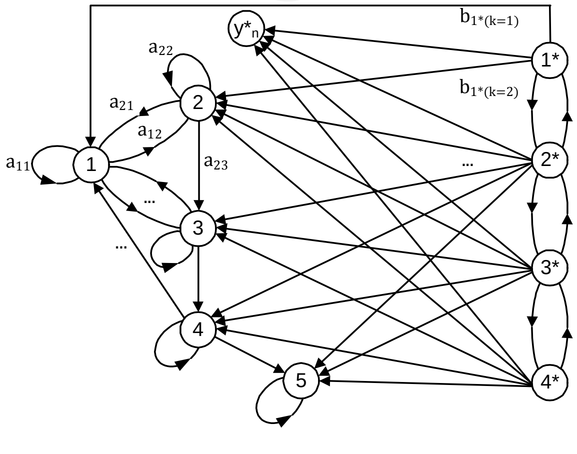


Figure 11. HMM of CI-risk

The alphabet Y_{AF} = *{SignificantHeartMurmur, Heart Rate>140,…}* covers the observables which are symptomatic for the occurrence of atrial fibrillation. The risk for cerebral insult is represented by the state space X_{CI}= *{No-CI-Risk, Low-CI-Risk, Medium-CI-Risk, High-CI-Risk}* and the alphabet Y_{CI} = *{{Cheynes-StokesBreath, Dizziness,…}* $\in X_{AF}$*}* including the different ranks of AF as an emission itself. According to the introductory described situation with an older person resting in his bedroom and AF as the point of interest, a critical situation may be identified by the repeated detection of a h*eartrate > 140*. The evaluation of this symbol over a deterministic period of time introduces a lot of interventions. The first step is the execution of the diagnosis assistant for the evaluation of the emission sequence to identify the best suitable model for the situation. This diagnosis assistant comprehends partial HMMs similar to the graph in figure 11. According to the decision of the assistant adequate actions will follow.

5. DISCUSSION OF THE RESULTS

The primal model for the position tracking via probabilistic techniques achieved only a detection rate of nearly *35% up to 37%.* After the calibration and adjustment of the transition and the emission matrices coefficients by the use of established algorithms like Baum-Welch the amount of recognized positions rose to an average of 85%. The important feature of self-control by learning is implemented and approved. The system is able to adjust its inference mechanisms with training data gathered from the process environment. The appropriation of the Viterbi-algorithm in the first implementation was very beneficial, referring to the reduction of computational load. In summary, the evaluation of the embedded home automation components based on HMMs is robust and meets the demands of the situation modeling, particularly of the emergency detection module which is a key feature for the proposed human centered assistance system.

The emergency detection module is one complementary part of the situation recognition. Therefore the approach to use the activity recognition like a feedback system for the reconfiguration of the information fusion systems regarding to health parameter threshold is very useful. Relating to this practice the rate of recognized contingency accomplished 72%, if the influence of state duration is considered. Without regarding to state duration, the detection rate dropped to modest 41%, the rate of misapprehension rose from 2 to 15%. The momentary weakness of the contactless measurement system for breath or heart rate detection over distance anticipates a better result, a mechanical feedback system between the assistance system itself and the patient like a biased-off switch is indispensable at the actual status of the research project.

In addition to the room of improvement for the sensor components and the sensor network topology, there is one critical point in respect to the evaluation of non-equidistant measurement data. This is, relating to trend analysis in medical context, an exacerbating threat for the modeling of the patients health status. Due to patient's behavior there is a temporal inconsistence in the measurement data particularly with regard to criteria like weight, blood-glucose or blood pressure. Coefficients of the emission matrices may be disrupted due to this fact and therefore some retaliatory actions relating to filter algorithms are required.

Unfortunately, time constants of medical or physiological processes are in general too big to determine the dependencies between different health states with learning algorithms at the assistance system. Another challenge is the interdependence between the diverse observables you want to analyze. Considering this fact, further information must follow from clinical randomized studies and must be used to determine appropriate stochastic models to expand the diagnosis assistance topology.

6. CONCLUSION

In this paper it was shown, that classified sequences of system states can be described as a statistical process by the use of Hidden Markov Models. For the use in the assistance systems, the system states are representing human behavior in the evaluation interval. The implemented methods are a useful tool to infer from uncertain knowledge, as long as stochastic independent parameters are being evaluated, the number of system states is kept low and appropriate training algorithms like Baum-Welch are used for the adjustments of the transition matrices. The integration of this functionality enables the assistance system to regulate systems with known properties as well as to evaluate system with unknown traits. The assistance system is able to make the necessary adjustments of its limited cognitive situation recognition by itself due to the learning process. For improving the diagnosis system it is necessary to evaluate existing and accessible data sets (Framingham Heart Study) and to obtain valid data sets by the use of randomized, multicentre clinical trials. The integration of time varying matrices for the transition and emission probabilities will be done in the next steps of the research project, so that the probabilistic modeling will be adapted on actual scenarios. Thereby the rhythm of user's life will be integrated in the assistance system also. The implementation of a_{ij} and $b_j(k)$ as *f(t)* should lead to an dynamic assignment of the coefficients. Another working package is the modeling and implementation of specific *ADL (Activity of Daily Life)* scenarios for different environments, which are based on valid data sets.

ACKNOWLEDGEMENT

This research is a part of the research project *'AAL@Home'* which is subsidized by the Bundesministerium für Bildung und Forschung in the context of *Ambient Assisted Living - AAL*.

REFERENCES

Book

Pearl, J., 1997. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., San Francisco, USA, ISBN 978-1558604797

Statistisches Bundesamt, 2009. *Bevölkerung Deutschlands bis 2060, 12. koordinierte Bevölkerungsvorausberechnung*. Statistisches Bundesamt, Wiesbaden, Germany

Klauber, J et al, 2010. *Krankenhaus -Report 2010–Krankenhausversorgung in der Krise ?*. Schattauer, F.K., 2010, ISBN 3794527267

Journal

Rabiner, R., 1989. A tutorial on hidden Markov models and selected applications in speech recognition, *Proceedings of the IEEE*, Vol. 77, No. 2, pp. 257–286

Hördt, M. et al, 2003. Differentialdiagnose und Dokumentation tachykarder Rhythmusstörungen, *Herzmedizin*, Vol. 20, No. 3, S. 146-152, 2003

Conference paper or contributed volume

Busch, B.H. et al a), 2010. Preventive diagnostics for cardiovascular diseases based on probabilistic methods and description Logic. *Proceedings of the 2010 IADIS European Conference on Data Mining 2010*. Freiburg, Germany, pp. 95-100. ISBN 978-972-8939-23-6

Busch, B.H. et al b), 2010. Architecture of an adaptive, human-centered assistance system. *Proceedings of the 2010 International Conference on Artificial Intelligence*. Las Vegas, USA, pp. 691-696. ISBN 1-60132-146-5

Helbig, M. et al, 2007. Ultrabreitband-Sensorik in der medizinischen Diagnostik. *Proceedings der 41. Jahrestagung der Deutschen Gesellschaft für Biomedizinische Technik BMT*, Aachen, Germany

Kouidi, E et al, 2006. Transtelephonic electrocardiographic monitoring for an outpatient cardiac rehabilitation programme. *Clin. Rehabil*.

Liao, L. et al,2005 . Location-Based Activity Recognition using Relational Markov Networks, *Proceedings of the International Joint*

Meyer-Delius D. et al, 2007, A Probabilistic Relational Model for Characterizing Situations in Dynamic Multi-Agent Systems. *In post-conference proceedings of the Conference of the German Classification Society - Gesellschaft für Klassifikation (GFKL)*, Freiburg, Germany

Subramanya, A. et al, 2006. Recognizing Activities and Spatial Context Using Wearable Sensors. *Proceedings of Conference on Uncertainty in AI (UAI)*. Cambridge, Massachusetts, USA

Taskar, B. et al, 2002. Discriminative probabilistic models for relational data. *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*. Edmonton, Alberta, Canada

Thomä, R. et al, 2007. UWB Sensor Networks for Position Location and Imaging of Objects and Environments. *Proceedings of the EuCap 2007*, Edinburgh, Scotland

Website

12. Koordinierte Bevölkerungsvorausberechnung, http://www.destatis.de/bevoelkerungspyramide/,(09/01/2010)

Copy Rights

FINDING THE TREE IN THE FOREST

Rikard König*, Ulf Johansson* and Lars Niklasson**
*School of Business and Informatics, University of Borås, Sweden
**Informatics Research Centre, University of Skövde, Sweden

ABSTRACT

Decision trees are often used for decision support since they are fast to train, easy to understand and deterministic; i.e., always create identical trees from the same training data. This property is, however, only inherent in the actual decision tree algorithm, nondeterministic techniques such as genetic programming could very well produce different trees with similar accuracy and complexity for each execution. Clearly, if more than one solution exists, it would be misleading to present a single tree to a decision maker. On the other hand, too many alternatives could not be handled manually, and would only lead to confusion. Hence, we argue for a method aimed at generating a suitable number of alternative decision trees with comparable accuracy and complexity. When too many alternative trees exist, they are grouped and representative accurate solutions are selected from each group. Using domain knowledge, a decision maker could then select a single best tree and, if required, be presented with a small set of similar solutions, in order to further improve his decisions. In this paper, a method for generating alternative decision trees is suggested and evaluated. All in all,four different techniques for selecting accurate representative trees from groups of similar solutions are presented. Experiments on 19 UCI data sets show that it often exist dozens of alternative trees, and that one of the evaluated techniques clearly outperforms all others for selecting accurate and representative models.

KEYWORDS

Decision support, decision trees, genetic programming, alternative solutions, inconsistency.

Copy Rights

1. INTRODUCTION

Decision support systems based on predictive modeling are today a crucial part of many organizations since data often is collected in amounts and with a complexity that exceed the capabilities of human decision makers. Even if accuracy normally is the main goal for predictive modeling Goodwin (2002) states that most decision makers would require at least a basic understanding of a predictive model to use it for decision support. Furthermore, Domingos (1977) points out that comprehensibility is important since it facilitates the process of interactive refinement that is at the heart of most successful applications.

One of the most popular techniques creating comprehensible models is decision trees algorithms such as C4.5 (Quinlan 1986). Decision trees are popular since they are fast to train and easy to understand. However, the trees still need to have a reasonable size to be considered comprehensible. Curruble et al. (1995) suggest that it becomes nearly impossible to get a global idea of a model if it consists of more than one or two dozens of rules.

A well known deficiency present in most decision trees algorithms is that they are unstable; i.e., slight variations in the training data can result in quite different attribute selections in the splits, see e.g., (Roiger & Geatz 2003). The problem then, of course, becomes which decision tree that should be trusted if small variations in the data will produce very different trees. Turney (1995) gives an example where engineers are disturbed and lose confidence in the decision trees when different batches of data from the same process result in radically different trees.

According to Dietterich (1996), the most fundamental source of instability is that the hypothesis space is too large. If an algorithm searches a very large hypothesis space and outputs a single hypothesis, then in the absence of huge amounts of training data, the algorithm will need to make many more or less arbitrary decisions, decisions which might be different if the training set were only slightly modified. This is called *informational instability*, instability caused by the lack of information.

This informational instability is for example often experienced in the medical domain, where datasets often contain small number of instances (sometimes 100 or less) but still relatively large number of features. In such cases (large spaces with sparse samples) it is quite likely that different logical expressions may accidently classify some data well, thus many data mining systems may find solutions which are precise but not meaningful according to experts; see e.g., (Grąbczewski & Duch 2002).

One approach to handle this problem is to instead benefit from the instability by creating a diverse ensemble of models using, for instance *bagging* (Breiman 1996) or *boosting* (Schapire 1990). However, even if these types of ensemble techniques most often are more accurate and stable in their predictions, they are not comprehensible since a large number of models would need to be interpreted to understand a prediction.

Another approach is taken by Grąbczewski & Duch (2002), who point out that many sets of rules with similar complexity and accuracy may exist, using for example different feature subsets, bringing more information of interest to the domain expert. Providing experts with several alternative descriptions make it easier for the experts to find interesting explanations compatible with their experience, and may lead to a better understanding of the problem. Consequently, data mining methods aimed at finding several different descriptions of the same relationship, are potentially valuable, and deserve investigation.

The approach of providing experts with alternative solutions is also supported by, for instance, Plish (1998), who notes that modern support systems for group decisions in situational centers largely depends on the availability of a procedure that generates "reasonable" (nearly optimal) alternative decisions in real time. If more than one solution exists, it would actually be misleading to present a single solution to a decision maker.

Following the argumentation above, this study will present a method for generating alternative solutions which all have comparable accuracy and complexity. Since too many alternatives risk to confuse a decision maker, the method also guides an expert among a large set of alternative solutions. In more detail, the suggested method first groups similar solutions and then selects accurate representative solutions from each group. Using domain knowledge, a decision maker could then select the best tree from only a few representative solutions. Finally, the decision maker could also request a small set of similar solutions to further improve his understanding of the relationship.

A solutions in the context of decision making could be any form, but this study only consider decision trees since they are one of the most popular machine learning techniques used for decision support. Solution and decision tree are used interchangeably in the rest of this paper.

Copy Rights

2. RELATED WORK

The following section will first describe some techniques for creating alternative solutions. In the last section different approaches for selecting a single model from several models will be discussed.

2.1 Generating Alternative Solutions

A straightforward and frequently used way of generating different solutions for a certain data set is to train different models on different parts of the training data. Since most machine learning techniques are instable, this will result in different solutions. Bagging is a well know example of this technique. In bagging, a new *bootstrap* training set is created for each model by randomly selecting instances (with replacement) from the original training set.

As mentioned above Pilsh (1998) acknowledge the importance of supplying alternative solutions to a decision maker. Pilsh also suggest an algorithm for generating alternative solutions for a multi-criterion linear programming model. Alternative solutions are generated by slightly altering either the objective or the constraints, and then solving the resulting problem with the help of supplementary constraints. Structural changes are also considered, but the algorithm is limited to multi-criterion linear problems, and cannot be applied to decision trees.

An algorithm for generating alternative decision trees is presented by Grąbczewski & Duch (2002), who use a variant of standard beam search to create heterogeneous forests of decision trees. The number of possible alternative solutions is restricted to the beam size. To create a forest, all trees that are found during the search are ordered according to their accuracy, estimated on validation set. An infinite beam size corresponds to a breadth first search, which is unpractical for most problems. Grąbczewski & Duch report that their algorithm finds several good alternative solutions for three UCI (Blake & Mertz 1998) data sets. However, it is somewhat unclear exactly how the solutions were evaluated, and the number of alternative solutions that were found is not reported.

Li & Lui (2003) present a simple method for creating ensembles called *cascading trees*. First all features are ranked according to their gain ratio. Next trees are created in a cascading manner where the root node of the i^{th} tree corresponds to the i^{th} ranked features; the rest of the tree is created as normal. The authors stress the fact that unlike bagging or boosting, cascading trees do not in any way modify the original data. This is of a critical concern in, for instance, bio-medical applications such as the understanding and diagnosis of a disease, where it is important that all training instances are classified correctly. The method is evaluated on two medical related dataset using 10-fold cross-validation. Several trees that could classify the training data without error were found for all folds. An interesting observation, made by the authors, is that the tree with the best test accuracy often did not have the feature with the highest gain ratio in its root.

It should be noted that ensemble creation techniques often create base classifiers by sacrificing accuracy for diversity. Ensemble members are usually less accurate than single model while the ensemble is more accurate. Hence, most ensemble methods are not suitable for creating several alternative standalone solutions.

In a previous study (Johansson et. al. 2010) we used GP to generate alternative models based on all data. Since GP is inherently inconsistent, no data needs to be scarified or modified to achieve alternative solutions. Several alternative trees where found and most trees were more accurate than a single decision tree created using CART (Breiman 1984).

2.2 Selecting Solutions

The most straightforward approach to selecting a single model from several alternative models, is of course, to compare all models and pick the *n* having the highest accuracy on either training data or on an additional (validation) data set. In this study however the starting point is models which all have comparable training accuracy which could complicate selection based on training accuracy.

Holding out a validation set is still applicable but previous work such as (Johansson et. al. 2010), has shown that even if a validation set is useful for selecting models, it also lowers the accuracy for the generated model since all data is not available for training. Again, the use of all available data for the actual modeling is especially important for data sets with relatively few instances to start with. Hence, selection based on validation accuracy will not be considered in this study.

A different approach is to select the *n* trees with the highest gain ratio, but as seen in the work of Li & Liu (2003) this does not yield very promising result.

In our previous study (Johansson et al. 2010), one tree was selected from a group of trees based on ensemble fidelity. The method used the fact that an ensemble of models most often is better than its individual members. First alternative trees were created from all training data using GP and all available training data. Next an imaginary ensemble was created from the evaluated models and used to generate predictions for both training and test instances. Finally, the predictions of each model were compared to the ensemble prediction and the model that was most faithful (in making the same predictions) was selected.

In the field of semi-supervised learning, this is referred to as *coaching*. Ensemble predictions could be produced even for the test instances, as long as the problem is one where predictions are made for sets of instances, rather than one instance at a time. Fortunately, in most real-world data mining projects, bulk predictions are made, and there is no shortage of unlabeled instances. Experiments using tenfold cross validation on 25 UCI data sets clearly showed that it is better to select models based on the fidelity against the imaginary ensemble than to use training or validation accuracy.

Copy Rights

3. METHOD

In this study we argue that alternative solutions may enrich a decision making situation. However, in the same way that complex decision trees become hard to comprehend, a large number of alternative solutions will also reduce the benefit of alternative solutions. Simply put, decision makers cannot interpret and evaluate dozens of trees, so there is a need for automatic strategies for selecting a subset of accurate trees.

We define an *alternative solution* as a decision tree that is of the same size or smaller than the original solution, while having equal or better training accuracy. Furthermore, an alternative decision tree should classify the data with a unique partitioning of the training instances, i.e., to be an alternative the solution needs to base its decision on different facts. It should be noted that two decision trees can classify the data in exactly the same way, but still partitioning the instances differently.

In this paper the *original solution* is represented by a decision tree created using the J48 algorithm in the WEKA (Witten & Frank 2005) workbench.

Naturally, the numbers of alternative solutions are dependent on the *size* of the tree, i.e. the number of *splits*, in the original tree and the number of possible attribute values in the data set. A *split* is a combination of an *attribute*, an *operator* and a *value* which creates a partition of the data set. In the experiments, (and in most decision tree algorithms), only *relevant splits* are considered, since the aim is to present truly alternative solutions. *Relevant* splits for an attribute is the splits that are needed to divide the data set into *pure* and *unpure* partitions. A *pure* partition is a set of instances of the same target class.

For an original tree with *n* splits and a dataset with *r* relevant splits there are n^r possible solutions. Of these solutions, only the ones with equal or higher accuracy are considered to be alternative solutions. Furthermore, if several alternative trees partition the data set in the same way, only the smallest tree is considered to be an alternative solution. Since the number of possible solutions is related to the size of the original solution, trees of the same size must be evaluated for all data sets.

3.1 Creation of Original Trees

Since the pruning in J48 does not support creation of trees of a certain size, the algorithm cannot be used in its original form. Instead a very large J48 tree is first created by setting the confidence factor to 0.5, (higher values yields warnings in WEKA). Next, the tree is pruned to a certain size in the following manner:

1. CUT_DEPTH is set to MAX_SIZE
2. All branches are cut at CUT_DEPTH and replaced with leaf node predicting the majority class of the training instance reaching the new leaf.
3. Redundant leaves are removed, i.e. if both leaves of any root split are predicting the same class the split is replaced by one of the leaves. This is done recursively since replacing a split can result in new redundancy higher up in the tree.
4. If the tree SIZE > MAX_ SIZE then CUT_DEPTH is set to = CUT_DEPTH-1. Return to 2.

Even if this pruning algorithm does not guarantee an exact size of the final tree, it is much more consistent than J48 original pruning algorithm.

3.2 Creation of Alternative Solutions

To be able to generate alternative solutions, a method needs to guarantee that the solutions have a certain training accuracy, a certain size and partition the data set in a unique way. None of the techniques discussed in the related work fulfill all three requirements.

Our approach is based on GP and continuously evolves a population of decision trees. If any of the trees in the population meets the requirement for being an alternative tree (accuracy, size and uniqueness) they are put in a growing list of alternative solutions. If two trees partition the dataset in the same way, only the smallest tree is kept in the list. To always drive the evolution towards new alternative solutions, a fitness function based on three metrics is used:

- A reward based on accuracy
- A punishment in relation to the tree size which increases if a tree is bigger than the original tree
- A punishment in relation to how similar the tree is to other alternative solutions.

Similarity is calculated by counting the number of identical splits that occurs in the same position in both trees. To ensure that each tree makes a unique partition of the data set, the GP is only allowed to search among relevant splits. If all splits are relevant, and the tree is not a copy of another tree, it will partition the data in a unique way.

The GP used in the experiments are more or less vanilla GP using tournament selection. A difference is that only two trees are selected for each tournament to slow down the convergence of the population. The idea is to look for more alternative trees in the neighborhood of discovered solutions. Another important

Copy Rights

difference from standard GP is that five batches are used in the experiments. A batch always starts from a new randomly generated population, but the list of alternative trees is kept during all batches. In this way, each population can start to look for solutions in new directions, even if a previous batch has converged to a certain solution.

Finally the trees are grouped based on their root split, putting all trees that start with the same split in the same group.

3.3 Selection of Representative Trees

As described in the related work, there are several ways of selecting a single tree from a group of trees. This study will evaluate four different techniques *random*, *training accuracy*, *ensemble fidelity* and *similarity*. *Random* selects a tree randomly and will be estimated by calculating the average accuracy of all trees. The other three techniques select one tree from each group of solutions. *Training accuracy* selects the tree with the highest training accuracy from each group. *Ensemble fidelity* is based on (Johansson et al. 2010) coaching technique and uses all solutions as an ensemble. The difference is that instead of selecting only one tree, the tree that is most faithful to the ensemble's test prediction is selected from each group. Finally, s*imilarity* selects the trees that have most in common with the other trees in the same group. The idea is that important splits will be used more often and hence the tree that has most in common with the other group members should contain more important splits.

4. EXPERIMENTS

The experiments are divided in three main stages, i.e., generation of alternative solutions, selection of representative trees and tree evaluation. All selection strategies were evaluated on the same groups of alternative solutions. It should be noted that only groups with three or more members were counted and evaluated in the experiments since *simlilarity* has no meaning for two trees. All experiments were performed on 19 data sets from the UCI–machine learning repository using 10-fold cross-validation with stratification.

In the experiments, J48 trees were created with 3, 5 and 7 splits. The values 3 and 7 were selected since they correspond to balanced trees. Trees of these sizes may seem small and simple but as pointed out by (Holte 1993), simple classification rules perform well for most common problems. Hence, small trees are a good starting point for a decision support system.

The GP process was implemented in G-REX, our publicly available GP-framework (König et al. 2008). In the experiments, a batch consisted of a population of 300 individual that were evolved during 100 generations with a crossover probability of 0.8 and a mutation probability of 0.001.

5. RESULTS

The result of the first experiments which concerns creation of alternative solutions is presented in Table 1 below. *#Inst* is the number of instances in the data set and *#Splits* is the number of relevant splits that each data set contains. *Size* is the actual size of the J48 tree after the second phase of pruning has been performed. The average number of alternative *trees* and *groups* (for ten folds) are presented for each of the evaluated tree sizes 3,5,7 splits (*3S, 5S, 7S*).

As seen in the table, a large amount of alternative solutions could be found for all target tree sizes. As could be expected, a larger tree size results in more alternative solutions. Ten or less alternative solutions could only be found for some data sets, i.e. seven for *3S*, four for *5S* and one for *7S*. The average number of solutions (81.5, 103.5 and 173.9) is obviously too large for a decision maker to handle manually.

Another interesting result is that when larger trees are allowed, fewer groups of similar trees are found, in spite of an increasing number of solutions.

Copy Rights

Table 1. Number of Solutions, Groups and average ACC

Data set	**#Inst.**	**#Splits**	**Size**			**#Trees**			**#Groups**		
			3S	**5S**	**7S**	**3S**	**5S**	**7S**	**3S**	**5S**	**7S**
Breast-cancer	286	32	2.6	4.2	6.1	64.3	90.3	159.9	6.7	5.4	3.6
Breast-w	699	74	2.0	5.0	5.3	113.0	325.4	302.4	12.2	13.8	12.9
Colic	368	311	2.0	3.5	5.7	4.6	24.7	95.7	0.0	0.8	2.5
C.-Lenses	24	4	2.9	2.9	2.9	8.2	7.4	7.4	1.6	1.4	1.3
Credit-a	690	860	2.5	4.0	6.0	26.0	106.8	237.2	1.9	3.4	2.7
Cylinderbands	540	1211	2.8	4.4	6.9	218.4	249.2	420.5	13.7	9.3	6.3
Diabetes	768	919	2.5	3.0	4.4	5.2	29.0	13.4	0.5	2.1	0.6
Glass	214	739	2.8	4.8	5.6	427.4	15.5	22.9	28.9	1.7	1.2
Haberman	306	80	2.4	3.9	6.4	142.5	324.0	551.9	12.1	17.2	20.1
Heart-c	303	316	2.6	4.5	5.5	76.9	17.3	34.1	3.5	2.0	2.1
Heart-Statlog	270	300	1.8	4.5	5.4	9.7	43.1	30.5	1.0	1.2	1.4
Hepatitis	155	201	1.9	3.6	5.3	19.0	145.9	280.2	1.6	6.1	6.5
Iris	150	59	2.9	3.8	3.8	28.5	21.8	23.8	4.5	2.7	3.2
Liver-disorder	345	274	2.1	4.6	5.6	66.7	106.2	88.1	5.8	7.6	3.8
Lymph	148	40	2.2	4.5	5.4	155.8	292.2	284.9	19.3	8.4	6.4
TAE	151	87	2.7	2.9	6.3	176.4	164.1	648.9	17.1	15.7	12.1
Tic-tac-toe	958	18	1.2	1.2	6.0	1.1	1.1	72.7	0.0	0.0	2.5
Wine	178	772	2.9	4.2	4.5	4.2	1.4	12.7	0.4	0.1	0.7
ZOO	101	96	3.0	5.0	6.8	2.7	1.9	16.4	0.1	0.2	0.7
MEAN	**350**	**336**	**2.4**	**3.9**	**5.5**	**81.6**	**103.5**	**173.9**	**6.9**	**5.2**	**4.8**

It should be noted that an alternative solutions is dependent on both the J48 size and accuracy. Hence, Table 2 below presents average test accuracy (*acc*) for each data set, tree *size*, number of alternative *trees* and *groups* with more than three trees. As could be expected, a larger J48 tree is more accurate than a smaller tree. It is, however, surprising that even if the accuracy increases, the possible number of solutions also increases. Of course a larger tree facilitates more combinations of the splits, but the search for an accurate tree also becomes harder since there are more trees to search among and less trees that actually have the required accuracy. Table 2 also shows that a possible explanation to the decreasing number of groups is increasing tree accuracy.

Copy Rights

Table 2. Size and acc vs. #trees and #groups

	Size	J48 acc	Rnd acc	#trees	#Groups
3S	2.4	73.5	75.1	81.6	6.9
5S	3.9	76.5	77.5	103.5	5.2
7S	5.5	76.6	78.2	173.9	4.8

The test accuracies for all selection techniques are presented in Table 3 where *Rnd* is the average accuracy of all alternative solutions, which represents selecting a tree at random. *Trn*, *Ens* and *Str* are the average accuracies for selecting a single solution from each group of solutions. *Trn* selects a solutions from the group based on training accuracy, *Ens* uses ensemble fidelity as described above and *Sim* selects solutions that is most similar (in terms of splits) to the other trees in the group. For each data set and tree size the best result is marked with bold numbers.

Table 3. Accuracy

	3S					5S					7S				
Data set	**J48**	**Rnd**	**Trn**	**Ens**	**Sim**	**J48**	**Rnd**	**Trn**	**Ens**	**Sim**	**J48**	**Rnd**	**Trn**	**Ens**	**Sim**
Breast-cancer	70.7	72.4	**73.0**	**73.0**	72.7	71.0	72.6	73.0	**73.3**	73.2	70.7	72.1	71.7	**73.5**	71.9
Breast-w	92.9	93.7	94.6	**95.0**	94.3	94.4	95.0	95.3	**95.4**	95.1	94.6	95.1	**95.4**	**95.4**	**95.4**
Colic	83.7	85.2	**85.9**	**85.9**	83.7	84.5	85.1	84.9	**85.3**	85.1	**85.1**	85.0	84.4	85.0	**85.1**
C.-Lenses	75.0	77.6	75.0	**78.3**	75.0	75.0	78.0	75.0	**81.7**	75.0	75.0	77.9	75.0	**78.3**	**78.3**
Credit-a	**85.7**	85.2	84.4	85.0	85.0	84.1	**84.8**	84.4	84.7	84.3	84.3	**84.8**	**84.8**	84.6	84.6
Cylinderbands	66.3	67.4	68.0	**68.5**	68.1	68.0	69.2	69.5	**69.8**	69.0	**70.9**	68.9	68.4	70.0	69.0
Diabetes	74.3	**74.4**	74.0	**74.4**	74.3	74.5	74.9	**75.1**	74.9	74.9	**74.6**	74.4	74.3	74.4	74.1
Glass	47.2	53.4	**56.5**	55.9	53.3	64.0	65.0	**65.7**	**65.7**	**65.7**	64.5	64.1	63.9	**64.8**	63.4
Haberman	68.3	72.4	72.2	**72.5**	72.1	69.0	72.1	72.9	**73.1**	72.2	67.7	72.5	72.2	72.7	**72.3**
Heart-c	75.2	74.4	74.0	**75.3**	74.9	78.2	79.9	79.9	79.7	**80.0**	77.2	80.8	81.0	**81.3**	80.7
Heart-Statlog	**72.2**	70.5	70.8	71.4	71.3	76.3	78.5	**79.6**	78.8	78.8	77.4	80.4	79.7	**81.9**	79.8
Hepatitis	**80.1**	78.8	78.3	78.2	77.7	78.8	81.3	**82.2**	81.9	81.6	76.2	77.9	78.2	**78.5**	**78.5**
Iris	94.7	95.0	94.9	**95.2**	94.8	94.0	95.2	**95.3**	95.0	95.2	94.0	95.4	94.7	94.8	**94.9**
Liver-disorder	64.4	66.5	66.5	**67.2**	66.2	65.0	64.4	**65.2**	64.7	65.0	**65.5**	64.5	63.7	64.5	63.6
Lymph	59.5	69.0	72.3	**72.4**	67.5	**78.3**	75.6	75.5	77.3	75.1	75.0	76.5	74.9	**76.8**	75.5
TAE	45.8	49.1	51.6	**54.0**	48.8	46.5	49.6	51.1	**53.0**	49.6	48.5	54.7	54.5	**55.7**	54.4
Tic-tac-toe	**68.8**	**68.8**	**68.8**	**68.8**	**68.8**	**68.8**	**68.8**	**68.8**	**68.8**	**68.8**	71.6	75.3	77.4	**77.7**	76.0
Wine	88.0	90.8	90.7	**91.4**	90.5	91.6	**92.9**	**92.2**	**92.2**	**92.2**	92.0	93.3	92.2	**94.6**	92.2
ZOO	83.3	81.5	**83.3**	**83.3**	**83.3**	**91.2**	90.2	90.2	90.2	90.2	91.1	93.1	94.1	**94.1**	94.1
Mean	**73.5**	**75.1**	**75.5**	**76.1**	**74.9**	**76.5**	**77.5**	**77.7**	**78.2**	**77.4**	**76.6**	**78.2**	**77.9**	**78.9**	**78.1**

For each experiment *Ens* achieves the highest overall accuracy, while the original J48 trees attain the lowest. It is also relevant to note that the alternative solutions generated with GP (Rnd) have a higher overall accuracy than J48. Since the other techniques select a subset of these solutions, they should also be better than *J48*.

Ens is clearly the best techniques with the highest overall accuracy for all experiments, and with the highest accuracy on 15 data sets for *3S*, 10 for *5S* and 11 for *7S.* A pairwise sign test at 0.05 significance level (presented in Table 4) shows that *Ens* is significantly better than all other techniques for *3S* and *7S.* For *5S* the results are not significant, but *Ens* clearly outperforms the other techniques and only loses three times agains J48, five times against *All*, six times against *Trn* and three times against *Sim*.

Table 4. Pairwise sign test

α=0.05	J48	All	Trn	Sim
3S Ens	**0.0125**	**0.0013**	**0.0042**	**0.0003**
5S Ens	**0.0075**	0.0963	0.6072	**0.0352**
7S Ens	**0.0192**	**0.0044**	**0.0013**	**0.0127**

Table 5. Average likeness of *Ens* trees

	Real # Ifs	Similarity Grp	Similarity All
3S	2.4	1.04 / 43%	0.27 / 11%
5S	4.0	1.94 / 48%	0.75 / 19%
7S	5.5	2.56 / 46%	1.02 / 19%

Finally, the group likeness of the trees selected by *Ens* is presented in the Table 5 above. The idea is that the selected trees should be accurate and representative for the group. To be representative the minimum requirement is that the solutions is more similar to the trees in the group than to trees not in the group. Since the groups are created from trees with the same root split, the similarity is at least 1.0 for any tree selected from a group (*Grp)* of trees.

Clearly the trees selected by *Ens* are more similar to the trees in the corresponding group than to all trees. Furthermore, the group similarity increases with size of the trees. On average the trees selected by *Ens* shares 46% of the splits with any member which should be compared to 16% shared with all alternative trees.

Copy Rights

6. CONCLUSION

In this paper we argue for a method aimed at generating a suitable number of alternative decision trees with comparable accuracy and complexity. When too many alternative trees exist, they are grouped and representative accurate solutions are selected from each group. Using domain knowledge, a decision maker could then select a single best tree and, if required, be presented with a small set of similar solutions, in order to further improve his decisions. The experiments support the feasibility of the purposed method since they show that:

- it is often possible to create many alternative trees, which all have comparable training accuracy and complexity. In average 120 alternative trees could be created for each original tree, which of course are more than what could be handled manually. Larger trees increase the number of alternative solutions even if the larger trees attain a higher accuracy.
- ensemble fidelity can be used to select several accurate trees from groups of alternative trees. In the experiment, trees were group based on their root split and an ensemble was created for each group. The trees that were most faithful to each ensemble (in terms of predictions) clearly outperform the average tree. Furthermore, trees selected in this manner are significantly better than the original tree and are also superior to selecting trees based on their training accuracy.
- the selected trees can be considered to be representative for their group, since they are more similar to trees inside than outside their group.

Copyrights

REFERENCES

Blake, C. & Merz, C., 1998. UCI repository of machine learning databases.

Breiman, L., 1996. Bagging predictors. *Machine Learning*, 24(2), 123-140.

Breiman, L., 1984. *Classification and regression trees*, Chapman & Hall/CRC.

Corruble, V., Thiré, F. & Ganascia, J., 1995. Comprehensible exploratory induction with decision graphs. *In Workshop on Machine Learning and Comprehensbility* (IJCAI).

Dietterich, T., 1996. Editorial. *Machine Learning*, 2(24), 1-3.

Domingos, P., 1997. Knowledge Acquisition from Examples Via Multiple Models. *In International Conference on Machine Learning*. Citeseer, p. 98–106.

Goodwin, P., 2002. Integrating management judgment and statistical methods to improve short-term forecasts. *Omega*, 30(2), 127–135.

Grąbczewski, K. & Duch, W., 2002. Heterogeneous Forests of Decision Trees. *Artificial Neural Networks (ICANN).*

Holte, R., 1993. Very simple classification rules perform well on most commonly used datasets. *Machine learning*, 11(1), 63–90.

Johansson, U., König, R., Löfström, T., Niklasson, L., 2010. Using Imaginary Ensembles to Select GP Classifiers. *In European Conference on Genetic Programming*, pp. 278-288

Johansson, U., König, R. & Niklasson, L., 2007. Inconsistency - Friend or Foe. *In 2007 International Joint Conference on Neural Networks*, pp. 1383-1388.

König, R., Johansson, U. & Niklasson, L., 2008. G-REX: A Versatile Framework for Evolutionary Data Mining. *International Conference on Data Mining Workshops*, 2008. ICDMW'08. p. 971–974.

Li, J. & Liu, H., 2003. Ensembles of cascading trees. *In International Conference on Data Mining (ICDM)*. pp. 585-588.

Plish, V., 1998. Algorithms generating alternative solutions for a multicriterion linear programming model. Cybernetics *and Systems Analysis*, 34(2), 301–304.

Quinlan, J.R., 1996. Bagging, boosting, and C4. 5. *In Proceedings of the National Conference on Artificial Intelligence*. p. 725–730.

Quinlan, J.R., 1986. Induction of decision trees. *Machine Learning*, 1(1), 81-106.

Roiger, R. & Geatz, M., 2003. *Data Mining: A tutorial-based primer*.

Schapire, R.E., 1990. The strength of weak learnability. *Machine Learning*, 5(2), 197-227.

Turney, P., 1995. Technical note: Bias and the quantification of stability. *Machine Learning*, 20(1-2), 23-33.

Witten, I. & Frank, E., 2005. *Data Mining: Practical machine learning tools and techniques*, Morgan Kaufman.

EVALUATING THE QUALITY OF SERVICE OF A DATA WAREHOUSING SYSTEM

Nélio Guimarães and Orlando Belo
Department of Informatics, School of Engineering, University of Minho - Campus de Gualtar, 4710-057 Braga, PORTUGAL

ABSTRACT

Data Warehousing Systems and Business Intelligence applications in general are some of the top priorities for many companies in the world. During their existence many of them have been spending large amounts of money in specific decision support architectures and related infrastructures in order to speed-up and scale-up their decision making services, and improve performance of their decision agents. However, in spite of being optimized systems – as expected – these expensive architectures not always provide fast and correct answers for ad-hoc queries, reporting services, or even fast processing times for multidimensional structures. A lot of reasons could explain such undesirable and not justified performance, ranging from complex cases of bad index structuring and join paths to simple overlapping of processes, running at the same time over the same analytical processor. As we know, such things are not simple to identify and solve in a convenient and practical way in effective decision-making scenarios. In order to sustain the quality of service of data warehousing systems we could use profiling and prediction strategies to identify and observe relevant system performance measures that provide us the means to establish better operational constraints. Based on this, we have designed and implemented a software tool especially oriented to predict and regulate the quality of service of a data warehousing systems, providing to their administrators the basis to establish more effective decision-making support services and the means to attenuate quality of service degradation.

Copy Rights

KEYWORDS

Data Warehousing Systems, Quality of Service, System Performance Indicators, Data Mining, Prediction Models.

1. INTRODUCTION

Today's decision agents are demanding for better systems, more reliable and efficient, at any level of service. As investments rise in decision support systems, and in analytical software in particular, decision agents also increase their demands for better *Quality of Service* (QoS). They are not so satisfied anymore with how analytical platforms are acting and responding to their needs. They expect that such tools present high levels of QoS. Daily, *Data Warehousing Systems* (DWS) face a large set of situations that constraint and degenerate the way in which decision support services perform and manage their tasks, provoking in many cases a lot of exploitation constraints to their users community. These cases use to decrease in a significant way systems' QoS, which affects consequently the service satisfaction of users. Commonly, DWS integrate in a same environment a lot of different technologies selected especially to provide to users – usually managers and top executive staff - the necessary knowledge and services oriented to support in an effective manner their decisions activities, contributing to improve their business performance and incomes (Vassiliadis et al., 1999).

During the last decade, several DWS development techniques have been proposed by many researchers (Kelly, 1997) (Kimball et al, 1998) (Bernardino et al, 2002) (Golfarelli & Rizzi, 2009) in order to improve system functionality, performance, reliability, and usability, just to name a few. However, as we know, DWS design and development are not simple. Current DWS design methodologies help a lot, but they don't solve it by themselves. There are several critical tasks in the entire development process, involving correct DWS application assessments, very demanding requirements analysis, as well a perfect notion about how decision support mechanisms and infrastructures will be implemented and explored. Additionally, for a medium-large company the resources involved (and their costs) are quite significant as well as are the risks they may have.

All of these issues reinforce, with no doubt, the need for high levels of information and service quality: two essential pears of the success of any DWS.

Today, DWS are recognised as crucial for any company, assuming tactical and strategic roles for daily business decisions. Their usage is continuously increasing as an important tool for decision support and an extreme valuable asset to deal with concurrency in real markets. In many companies, analytical applications need to deal with large volumes of information, many of the time pre-materialized in very sophisticated structures of aggregated data, and intensive ad hoc querying, which impose high performance levels to DWS servers. DWS users expect a quite response for any of their queries, independently for their complexity and data volumes involved with. Sometimes, it’s not simple to maintain a good QoS in this kind of platforms. There are times where users launch an enormous number of queries, without warning, because simply they need to do it right now. A poor QoS could cause a very strong impact in users’ confidence. When they are waiting more than they consider acceptable for an answer to a query, they use to apply to other alternatives that provide them the information they want in the time they expect. Usually, users do not care about the reasons why system performs so badly. Even when they are quite reasonable and justified, technical aspects do not attract user attention or attenuate their disappointment. Due to its importance, in many cases, this issue has triggered expensive investments in complex analytical systems in order to improve query responses and processing time when systems are overloaded with intensive ad hoc querying. Nevertheless, the return-on-investment, in terms of time and money, should be considered when a less expensive solution could be found and implemented. In fact, any service provided by a system, at any time, must return a good QoS. The same must happen with a DWS. During the last few years, researchers had given a lot of attention to this area, making efforts in the way to discover and develop new methods, strategies and systems that contribute to improve QoS in specific working areas (Kavimandan & Gokhale, 2007) (Juanole & Mouney, 2006). QoS is quite determinant in the adoption and success of any computational systems. DWS are no exception (Li et al., 2007) (Thiele et al., 2007) (Costa & Furtado, 2007).

In order to contribute to attenuate the effect of low rates of QoS in DWS, we designed and implemented a non-intrusive system - DWQoS-PP - with the ability to load essential QoS measures (exclusively related to querying output) and to predict the performance of a DWS for a specific period of time. With the information provided by the system it is possible to discover and discuss potential periods of QoS deterioration as well to suggest a new execution order for daily decision support tasks for such periods, attenuating its impact on decision support systems’ performance and consequently on their users’ satisfaction. This paper is organized as follows: section 2 presents the DWQoS-PP system’s architecture, the designed QoS model and QoS classes, and the performance prediction model used to establish the QoS deterioration periods; section 3 contains a detailed evaluation and discussion of the results obtained during system validation and testing; finally, section 4 presents some concluding remarks and some research lines for future work.

2. THE DWQOS-PP SYSTEM

2.1 System’s Architecture and Life Cycle

The DWQoS-PP system was designed and implemented (Figure 1) as a non-intrusive system with the ability to act transparently as a monitoring agent over the queries that are sent to a target DWS by its users. It has the ability to analyze systematically its behaviour, examining periodically some pre selected key values related to querying performance, trying to detect anomalous QoS situations based on a set of heuristics and operational status values. The system’s way of acting – life cycle - is not complicated. Every time a user submits a query to the DWS a profiler event is triggered. This creates a new record on a temporary table, used only by the profiler, with some values related to the query execution (e.g. duration, or number of writes). At the same time, a daemon collects continuously other important operating system performance variables (e.g. number of current processes, percentage of CPU usage, or memory usage). Latter, all the information gathered by those two collectors programs is conciliated on a *Data Staging Area* (DSA) to be clean and organized accordingly to the QoS dimensional models stored in the DWQoS. Finally, the prediction mechanisms are activated in order to generate potential future QoS deterioration periods. Basically, this is the DWQoS-PP life cycle.

Copyright

The DWQoS-PP's architecture integrates several functional blocks that were defined based on a typical DWS platform. They involve an information source, a data staging area, and a data warehouse repository. Additionally, a set of services was implemented to gather queries requests from decision makers, operational systems functioning metrics from operational systems, and prediction models and mechanisms to generate queries execution plans. System's functional blocks are the following:

– A query request generator, which simulates all the possible requests and data interaction between the target DWS and the system's users (or applications). This system's component is quite flexible having the possibility to adapt its query generation process to all types of requests, from conventional human interaction to applications requests.

– A profiler that is an application that can be found today in the most common data base management systems. This module can be seen as an application with the responsibility to record continuously QoS measures that were selected for monitoring - all queries submitted to the target system are analysed. The profiler may have several configurations in order to receive different monitoring plans, each one corresponding to a new set of operational variables. Thus we can monitor QoS measures directly related with the DBMS execution, such as DURATION or N_WRITES, among others. It's important to note that we did not analyze the queries themselves but only the correspondent system's behaviour.

– A software daemon, which has the responsibility to monitor the operating system key performance variables. In the current version, this daemon is a system's script that every 30 seconds records the number of current processes (N_PROC), the percentage of CPU usage (%_CPU), and the memory usage (MEM_U) – these were the variables that we defined as the necessary (and most appropriated) to fulfil prediction's requirements and our current system's goals.

– A performance prediction module (the most important module of the system) that has the ability to classify all the queries into five possible classes of QoS. The QoS classification method that was used is a result of a composition of several data mining techniques, namely: clustering, classification and linear regression.

– A DWeb that is the target data warehouse on which the QoS system was built. It is a specific type of a data warehouse, which store information in a dimensional way about web navigation.

– A data staging area (DSA), which supports all system's populating jobs, conciliating accordingly the information provided by the system's daemon and profiler.

– The DWQoS is the system's data repository (a typical data warehouse) where we store, in a dimensional way (Kimball & Ross, 2002), the data that will be used by the performance prediction model in order to generate the QoS index for the target DWS.

Copy Rights

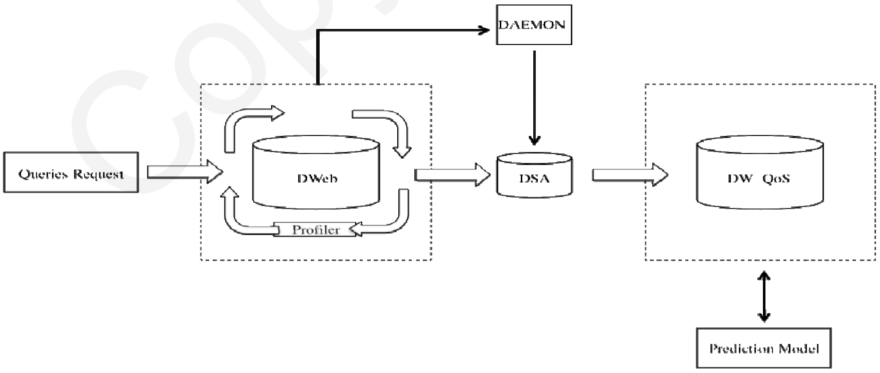


Figure 1. The DWQoS-PP System's architecture

The DWQoS dimensional structures - integrated in a star-schema that has a bridge table (Figure 2) - were designed taking into consideration the specificities of the prediction services, providing a valid context, quite useful and consistent, where facts about query execution and performance metrics are stored. The grain of the base fact table corresponds to the data about a simple query, involving one or more DWS's tables. This grain definition gives us the essential information to get historical execution plans and predict what can be done in future to avoid (or at least attenuate) performance bottlenecks or high constraining ad hoc queries. This is not very difficult to get once we can easily obtain strong and supported information in the fact table about the most high demanding periods, the execution time for the most frequent queries, or what are the most requested tables, just to name a few.

The DWS base schema includes five dimensions, each one corresponding to an axis of analysis defined by us, in order to support QoS monitoring and classification. The dimensions are: 'Calendar', which integrates, as usual, a real-world calendar; 'Time' that contains hours, minutes and periods of a day; 'User_U', which provides information about the users who launched the queries; 'Query', the queries received in the system; and, finally, 'Table_T', that stores information about the tables involved in the queries. With respect to the fact table, it integrates the usual dimensional attributes corresponding each one to one of the selected dimensions and set of relevant measures (Table 1), basically a set of performance indicators. Latter, in the prediction module, these attributes will be complemented with new ones – the goal attributes – that will receive the results of the prediction processes.

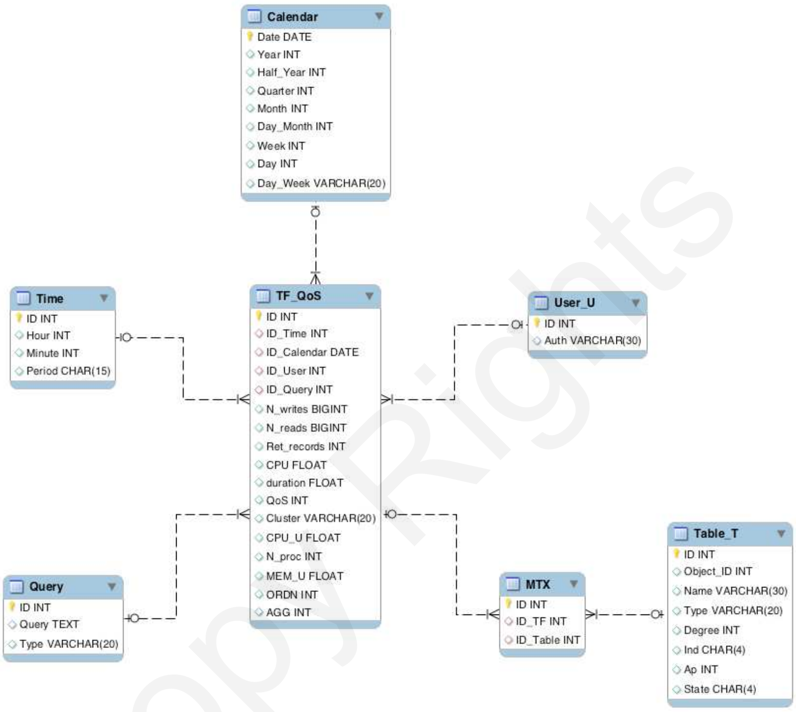


Figure 2. DWQoS-PP System's data model

Copy Rights

2.2 Performance Prediction Model and Classes

Based on the request classification proposed in (Kang, 2003) and (Kang et al, 2004), we designed a query classification model supported by five classes of service 1, 2, 3, 4, and 5, according to their importance, where class 5 corresponds to the higher level of quality-of-service. Every class deals exactly with the same QoS measure (Table 1) and classifies each cluster in a qualitative way. The QoS classes are created from the training data set just only once. However, they could (and should) be refreshed every time the system suffers significant changes that have direct impact on the correspondent QoS level, i.e., it may be important to distinguish different types of queries that belong to a same previous class. The data set is composed only by the QoS measures that are presented in Table 1. Basically these measures are only a sub set of all the possible measures that we can collect in the system's fact table. Using a cluster algorithm we create an output for five clusters, corresponding each one to a different QoS class.

Table 1. Selected DW QoS measures.

QoS Measure	Description	System Module
N_WRITES	Number of physical disk writes performed by the DBMS during the query execution.	Profiler
N_READS	Number of logical disk reads performed by the DBMS during query execution.	Profiler
RET_RECORDS	Number of returned rows.	Profiler
DURATION	Time since the query is submitted until it finished.	Profiler
%_CPU	CPU usage during the query execution (note that this measure does not record only the CPU usage for the system event triggered by the query; the value returns all CPU usage).	Daemon
N_PROC	Number of process running during the query execution.	Daemon
MEM_U	Total of memory usage during query execution (note that this measure does not record only the memory used for the system event triggered by the query).	Daemon
AGG	If the query has a *group by* clause the value is 1 otherwise is 0.	Query Parsing
ORDN	If the query has a *order by* clause the value is 1 otherwise is 0.	Query Parsing

3. EXPERIMENTAL EVALUATION AND RESULTS

The DWQoS-PP was implemented entirely in Java, integrating four distinct packages involving a set of twenty-four different classes. The target DWS that was selected to support this study was installed on a Microsoft SQL Server 2008 (MSSQLS) DBMS on an Intel Pentium IV 3.4 GHz platform, with 1GB of RAM. We used the default MSSQLS installation and configuration without any kind of performance optimization. The DSA and the DWQoS are both located on a MySQL 6.0 DBMS. The data warehouse used to evaluate the system received all the clickstream data gathered on each Web site under monitoring. Most of the queries sent to the system come from different computer applications, like OLAP platforms, reporting services applications, and other sporadic applications selected without any special criteria. However, a minority percentage of the entire set of queries comes from ad-hoc requests. Figure 2 gives a general idea about the query density by hour on a normal working day.

Copy Rights

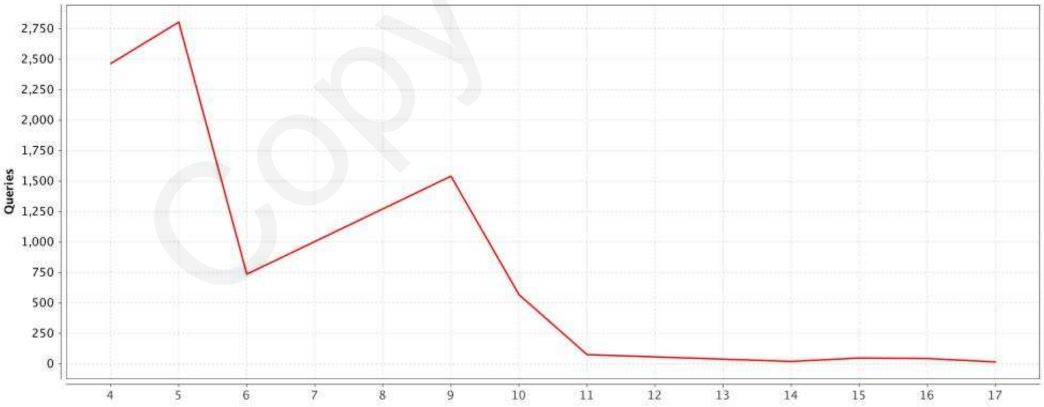


Figure 3. Query distribution (aggregated by hour of a working day)

As we can see in Figure 3, there were three distinct time periods of great query intensity - the first one occurred between 4 and 5am, the second one at 6am, and, finally, the third period at 9am. The first period corresponds to the moment of processing an OLAP cube. The second one was the time when a reporting server refreshed all the users' reports under its supervision. The last period is related to a normal workday routine and some more other reporting services tasks. The creation of the clusters was an important step for the QoS class construction. They were generated using the EMClustering algorithm (Dempster et al. 1977) (Sundberg 1974). This is one unsupervised learning algorithm that follows a simple and easy way to classify a given data set in a certain number of clusters (we assumed five distinct clusters, fixed a priori). The main idea was to define *k centroids*, one for each cluster, and associate each point belonging to the data set to the nearest *centroid*. Assuming this principle, we created five clusters with 164, 1076, 352, 5944, 780 items, respectively (Table 2). As one knows, users are an important source of information. Thus, we decided to collect some selected inputs from them during some regular working sessions with the system. The knowledge acquired was quite relevant and told us that the system has a positive overall assessment. In fact, 62% of the users evaluated the system as good, 17 % evaluated it as satisfactory, and the remaining 21% were positioned between a very good and a bad appreciation. Note that no one had classified the system as very bad, when encouraged to evaluate the system as very good, good, satisfactory, bad and very bad. With this information we built a decision tree for the classification of the clusters (Figure 4).

Table 2. An Example of the average values for 'Cluster0' and 'Cluster1'

Cluster	Attributes	Average	Items
Cluster 0	N_Writes	0.073	164
	N_Reads	11,433.585	
	Ret_Records	850,725.561	
	Duration	35.522	
	CPU_U	74.756	
	N_Proc	57.488	
	MEM_U	882.553	
	ORDN	0.024	
	AGG	0.073	
Cluster 1	N_Writes	44.532	1076
	N_Reads	134,824.439	
	Ret_Records	6,447.506	
	Duration	20.121	
	CPU_U	16.672	
	N_Proc	54.1	
	MEM_U	899.313	
	ORDN	0.896	
	AGG	0.948	

Copy Rights

To build the decision tree we assumed (based on personal experience with this kind of systems) that there are some system's operational measures that affect more than others the quality of service of a system in the satisfaction of a query. Thus, we sorted the decision parameters in the decision tree based on what is more discriminating to what is less discriminating. To classify the clusters we have used the overall average for each attribute in the data set as a point of decision, and for each cluster, using their attributes average, we decide if the attribute is greater or less then the overall average of the data set, and so on. It's important to note that we do not use a regression classification tree because to use this type of technique it is necessary to have a target attribute on the training data set already classified. So, by not having such target attribute classified we used this kind of variation for a decision tree. The final classification was: 'Cluster 0' – Satisfactory, 'Cluster 1' – Satisfactory, 'Cluster 2' – Bad, 'Cluster 3' – Good, and 'Cluster 4' – Very Good. Using the QoS class classification, the previous clusters were classified as 'Cluster 0' – 3, 'Cluster 1' – 3, 'Cluster 2' – 2, 'Cluster 3' – 4, and 'Cluster 4' – 5. With this defined it's now much more easy to predict system's querying performance. First we updated the QoS and clustering attributes from the system's fact table with the corresponding values, and then we applied a linear regression algorithm [10, 11] to create a linear equation in order to allow us to calculate, finally, our proposal for a QoS index. After submitted the data set to the linear regression algorithm (Breiman et al, 1984) we got the following equation to get QoS index:

```
QoS ← 0.007 * N_writes - 0.011 * Duration + 0.022 * CPU_U
                       + 0.924 * N_proc - 0.005 * MEM_U + 21.624
```

Root means squared error: 4.960 +/- 0.163 (mikro: 4.963 +/- 0.000).
Absolute error: 3.361 +/- 0.134 (mikro: 3.361 +/- 3.651).
Relative error: 5.11% +/- 0.20% (mikro: 5.11% +/- 5.35%).
Correlation: 0.724 +/- 0.012 (mikro: 0.723).

Finally, we calculate the overall DWS QoS value, which required only an average of all QoS queries. The result for the target system was approximately of 63.4%.

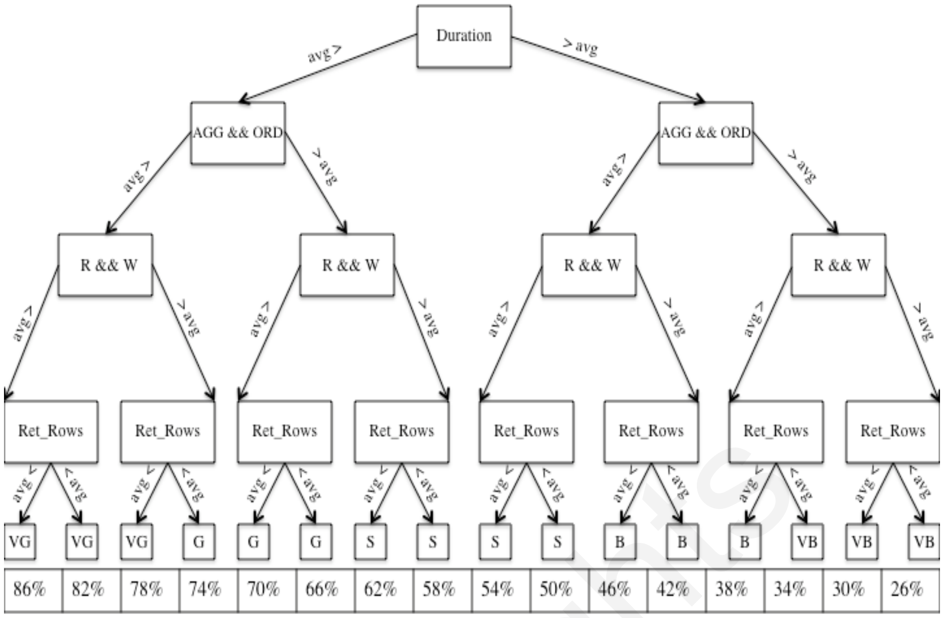


Figure 4. Decision tree for clustering support

Copy Rights

4. CONCLUSIONS AND FUTURE WORK

In this work a non-intrusive QoS performance prediction system for DWS was presented and discussed. It was a first attempt to make an effective QoS evaluation for a specific DWS in a predefined time period. We believe this first successful effort could be especially useful on every DWS that has a large querying density, a large numbers of ad hoc queries per day. This work demonstrated that it is possible to monitor and control the QoS of a DWS, appealing for quite inexpensive computational means and using up-to-date technologies. Its flexible design and modular architecture allows the system implemented to be extensible to other DBMS beyond the one used as the main data storage platform. However, at the moment, the system is only prepared to monitor and make prediction analysis for DWS deployed over Microsoft SQL Server platforms. Nevertheless, in a near future, we intend to prepare the system to work over the most important DBMS available on today’s databases software market.

Using clustering and numeric prediction data mining techniques provided us a solid background to establish effective querying profiling in order to classify accordingly the metrics selected for QoS evaluation, as well gave us the possibility to define a regression equation with the ability to make the calculation of a satisfaction index for a target DWS: our primary goal. Such kind of index could be a first approach to have the possibility to monitoring continuously the behaviour of DWS, and every time the index reach values considered critical, it is possible to know what were the functional parameters that contributed to trigger those strange values. Knowing that, a DWS administrator expert can identify the causes of such performance deterioration acting accordingly. However, having already a useful tool to manage DWS QoS does not mean that our work finished here. These were only the first steps towards the establishment of a model to predict effectivelly service degradation in DWS environments.

REFERENCES

Bernardino, J., Furtado, P, and Madeira, H. (2002). DWS-AQA: A Cost Effective Ap-proach for Very Large Data Warehouses”. IDEAS 2002: 233-242.

Breiman, L., J. H. Friedman, R. A. Olshen, and C. J. Stone (1984). Classification And Regression Trees. New York: Chapman and Hall.

Costa, R., Furtado, P., "An SLA-Enabled Grid DataWarehouse," Database Engineering and Applications Symposium, 2007. IDEAS 2007. 11th International , vol., no., pp.285-289, 6-8 Sept. 2007

Dempster, A., Laird, N., Rubin, D. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. Journal of the Royal Statistical Society. Series B (Methodological) 39 (1): 1–38.

Golfarelli, M., Rizzi, S., Data Warehouse Design: Modern Principles and Methodologies, McGraw-Hill Osborne Media; 1st Edition, 2009.

Juanole, G., Mouney, G., Real Time Distributed Systems: QoS and Impact on the Performances of Process Control Applications, In Proceedings of the 17th International Symposium on Mathematical Theory of Networks and Systems, Kyoto, Japan, July 24-28, 2006.

Kang, K., D., “QoS-Aware Real-Time Management”, PhD thesis, U. Virginia, May 2003.

Kang K. D., S. H. Son, J. A. Stankovic, “Managing Deadline Miss Ratio and Sensor Data Freshness in Real-Time Databases” IEEE Trans. on Knowledge and Data Engineering, Vol 16, Nr 10, Pages 1200-1216. October 2004.

Kavimandan, A. and Gokhale, A. Supporting systems QoS design and evolution through model transformations. In Companion To the 22nd ACM SIGPLAN Conference on Object-Oriented Programming Systems and Applications Companion, Montreal, Quebec, Canada, October 21 - 25, 2007.

Kelly, S., Data Warehousing in Action. John Wiley & Sons, Inc., New York, NY, USA, 1997.

Kimball, R., Reeves, L., Thornthwaite, W., Ross, M., The Data Warehouse Lifecycle Toolkit: Expert Methods for Designing, Developing and Deploying Data Warehouses with CD Rom. John Wiley & Sons, Inc., New York, NY, USA, 1998.

Kimball, R., Ross, M., The Data Warehouse Toolkit: The Complete Guide to Dimensio-nal Modeling, Wiley; 2nd ed, April 26, 2002.

Li, W., Gao, D., Bhatti, R., Narang, I., Matsuzawa, H., Numao, M., Ohkawa, M., and Fukuda, T. Deadline and QoS aware data warehouse. In Proceedings of the 33rd international Conference on Very Large Data Bases (Vienna, Austria, September 23 - 27, 2007). Very Large Data Bases. VLDB Endowment, 1418-1421, 2007.

Sundberg, R. (1974). Maximum likelihood theory for incomplete data from an exponential family. Scandinavian Journal of Statistics 1 (2): 49–58.

Thiele, M., Fischer, U., and Lehner, W. Partition-based workload scheduling in living data warehouse environments. In Proceedings of the ACM Tenth international Workshop on Data Warehousing and OLAP (Lisbon, Portugal, November 09 - 09, 2007). DOLAP '07. ACM, New York, NY, 57-64, 2007.

Vassiliadis, P., Bouzeghoub, M., Quix, C., Towards quality-oriented data warehouse usa-ge and evolution. In CAiSE ’99: Proceedings of the 11th International Conference on Ad-vanced Information Systems Engineering, pages 164–179, London, UK, 1999. Springer-Verlag., 1999.

Copy Rights

ACCESS TO RELATIONAL DATABASES IN NATURAL LANGUAGE

Nikos Papadakis*, Pavlos Kefalas**and Aris Apostolakis***
*Department of Sciences Technological Educational Institute of Crete, Heraklion, GR 71004**
*Department of Informatics Aristotle University of Thessalonica***
*Department of Computer Science University of Crete****

ABSTRACT

In this paper we propose a tool which has the ability to recognizes a subset o English language and translate them in to the corresponding SQL command. The system receives as input a query in a well specified subset of the English language and a database scheme. It processes this data accordingly to certain rules and acknowledgements in order to produce as output a correct query in sql language paper must have an abstract.

KEYWORDS

Natural Languages Process, AI in Databases.

1. INTRODUCTION

The natural language question answering systems are aspired to become the next generation in the communication between user and computer. They allow users to interact with databases without them having specialized knowledge over query languages. The purpose of this paper is to provide an interface in natural language to databases that will accommodate users that interact with databases and in particular regarding part of querying them.

Because the information retrieval from relational databases requires certain knowledge of specific programming languages (e.g. SQL), there is an obvious need to create a system that will allow the user to post queries, as in their everyday life, phrased in natural language.

It must be stated that those needs were noted since the decade of 1960. The ISNLIS (LUNAR SCIENCE NATURAL LANGUAGE INFORMATION SYSTEM) was one of the first experimental question answering systems, which allowed geologists to have access, analyze and compare data obtained from the analysis of rocks that the Apollo missions had taken from the moon. It is obvious that the lack of such a system would render compulsory the need to teach the appropriate programming language to the geologists, which would have serious consequences in both cost and time.

In this paper we have set as purpose the construction of a system that will receive as input user queries to a database expressed in natural language and will produce as result these queries in sql language. For the achievement of this purpose we created a grammar that is based on the syntax of the imperative in English and on the method of query editing in sql language. Substantially, we created a query language very close to natural language, which gives the ability to unskilled users to query a database.

For the construction of the system we used lexicographical and syntax analysis of language techniques as regards to the processing of the query that the user enters, artificial intelligence techniques for the production of a sound sql query while using in parallel knowledge on databases (relations between tables of the database, keys of the tables etc). Finally for the construction of the system we used the C programming language.

Our system, besides the processing of the query that the user enters, also processes the database scheme. This means that the system can have as input any database scheme (independent from Database scheme) and in addition, the query processing has direct relevance to the database scheme.

The user of the system must keep in mind the limitations that we have set, in order to create a query and must also have knowledge of the elements of the database is used.

Copy Rights

The rest of the paper is organized as follows: in section 2 we present the most important previous works, while in section 3 we present our systems (architectures and examples of its operations.) Final in section 4 we conclusion, and describe possible extension of or systems. In appendix A we give the grammar of Natural Language which we use as interface to the database.

2. PREVIOUS WORK

The first database interaction systems in natural language appeared in the 70's, when the use of databases began to grow. One of the most familiar systems was the LUNAR[Woods W. et. al. 1972], which was presented in 72' and provided interaction with a database containing information concerning rocks that the missions to moon brought back. It was based on syntax analysis and was able to create multiple analysis trees for the same query. It proved ineffective because it was very specialized. The LADDER [Hendri G. et. al. *1978*] was the first system to include grammar analysis, communicating with a database that contains information concerning the U.S. navy ships. The most familiar example of database interaction system in natural language is the CHAT-80 [Warren D. and Pereira F. 1982]. The CHAT-80 communicates with a database that contains geographic information. The code of this system is free and is nowadays still in use. The CHAT-80 converted queries in English to prolog plans, which were resolved with the use of a database. Its spread was wide and it was the basis for the development of more experimental systems, like the MASQUE [Androutsopoulos I. et. al]. Similarly operates the PRECISE [http:// 216.239.59.104/ search?q=cache:QxnIsr9UPwkJ:seattleweb.intelresearch.net/seminars/presentations/ Feb19_03_OEtzioni/ TalkingToYourToaster.ppt+precise+nlidb&hl=el&ct=clnk&cd=4&gl=gr&client=firefox-a], which in case it can not give an answer, indicates which part of the query it could not understand.

Latter on was the RENDEZVOUS [E.F. Codd 1974] created, which initiated a dialogue with the user in order to help them form the query. In the middle 80's, the database interaction systems were a research object and various systems had been created. For example, the TEAM [Grosz J. et. al. 1987] was created for database administrators lacking great experience. The ASK [Thompson B. and Thompson F. 1985] provided to the users the ability to teach the system new words and meanings, throughout the duration of the dialogue. It had the capability to communicate with external databases, such as e-mail management programs and other similar applications. All the applications were available to the user through natural language interaction.

Through the progress of time, similar systems that used other languages beyond English were developed, such as [Shan Wang et. al. 1999], which is a database query system that receives queries in the Chinese language.

Finally, a relative paper [Seymour Knowles 1991] took place at the university of Canterbury in 1999, which demonstrated an information recovery system from a database that received queries in natural language.

3. ARCHITECTURE AND IMPLEMENTATION OF OUR SYSTEM

In this section, we present the architecture and the functionality of the tool. Figure 1 illustrates the architecture of our tool as a component diagram.

3.1 Architecture of the System

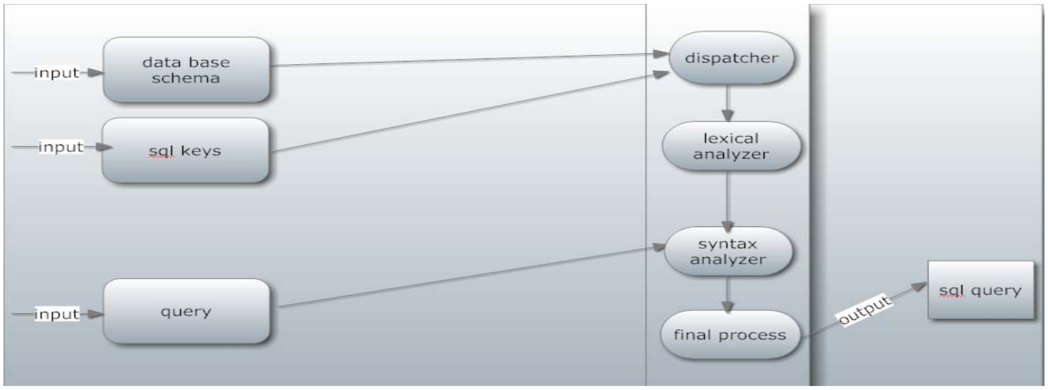


Figure 1. The architecture of our system.

The tool consists of a set of intercommunicating components with well-defined roles. The parts that our system is comprised of are:

Data base schema: The file that contains the scheme of the database.

Sql keys: The file that contains the relation of words-phrases to sql keywords.

Query: The file that contains the query that the user enters.

Dispatcher: The processing stage of the data base schema, sql keys and the code production stage for the file that will construct the lexical analyzer.

Lexical analyzer: Execution of the lexical analyzer with the query file that was given.

Syntax analyzer: Execution of the syntax analyzer with the lexical data from the lexical analyzer and the grammar.

Final process: Processing of the results of the syntax analyzer accordingly to certain rules.

Sql query: Printing of the final sql query.

Our system follows the following processing stages:

A) As initial input, the system receives and processes the files that contain the database scheme and the keywords. The processing that takes place is to store them to relating structures and then, based on them, the code for the file that will construct the lexical analyzer is produced.

B) As second input we receive the file that contains the user's query. The query undergoes 2 processing stages. The first is the lexical analyzer, which reads and converts character sequences of the input file to certain lexicographical elements. For this stage we have used the lex tool(Lex is a generator of lexical analyzers for the C and C++ languages). The second stage receives these lexicographical elements as input and, based on the grammar that we have defined, produces sql code. In this stage we have used the bison[1] tool for the production of the syntax analyzer.

C) In this stage, the code that was produced by the syntax analyzer is processed according to specific rules and the final sql query is produced, which is then displayed on the screen.

ures should de numbered consecutively as they appear in the text.

3.2 System Input Analysis

Our system receives 3 text files as input. The database scheme that defines which database the queries are applied on. The file with the relations of words-phrases to sql keywords that the user gives, which can also be filled in, as will be stated below. Finally the user's query that is entered every time the system is run. We represent analytically the syntax method of the three input files so that the system will produce correct results.

Copy Rights

[1] Bison tool: Bison is a generator of syntax analyzers for the C and C++ languages. It converts the description of a context free grammar to a LALR (Look Ahead Left to right parse Right most-derivation) syntax analyzer.

a. Database scheme

In order for our system to work and produce a sql query, it must know the database scheme on which the processing of the user's query will take place. The database scheme must be provided by the user. Namely, our system expects the user to enter a text file named "data base.txt". The file must have a certain syntax form stated below:

The table's name must obligatorily be first and followed by the ":" punctuation mark.

Then the table's elements are written, followed by a comma. (,)

If an element is a key of the table, it must be followed by a Greek question mark. (;)

If it is a foreign key, it must be followed by an English question mark. (?)

Besides the table's name, the rest elements have no restrictions on the input order. For example, either the following table is inserted

customers:cname,cid;,city,aid?,

or

customers: cid;,city,cname,aid?,

is the same for our system.

The system can receive any database scheme. The only restriction is databases containing tables that are not defined by a unique key (that is tables having 2 or more of their elements as keys).

An example of an acceptable database is:

```
customers:cname,cid;,city,aid?,
agents:aname,aid;,acity,
orders:orid;,month,cid?,aid?,pid?,
products:pid;,price,
```

b. relation of words-phrases to sql keywords

The system works for simple sql queries of the form *select (distinct) from (where)*. Namely, from the sql word-phrase set we work with only the 4 aforementioned words. The system relates the 4 aforementioned sql keywords to English words that are provided to the system through a text file (sqlkeys.txt). The relation that we provide is for the *select* the *find* word, for *from* the words *from* and *of*, for *distinct* the word *unique* and for *where* the word *that*. This convention is not given by the user and is taken for granted by the system. Of course, it is can be easily expanded, offering a greater range of words-phrases that relate to keywords. This is analyzed thoroughly below, at the *code expandability* chapter.

The file that we use in our system is:

```
find:select
of:from
that:where
unique:distinct
```

c. user query

As stated above, our system supports only simple sql queries of the form *select (distinct) from (where)*. In order to have a query accepted by the system, certain preconditions and limitations on the syntax of the sentences must be kept. Specifically:

In our system there is no distinction between uppercase and lowercase letters. For example, the word *find* treated as the word *FiNd*.

The user, in order to refer to an element of a table of the database must write it with the same name mentioned in the database. In order to refer to a table, the user must state the name of the table exactly as it is mentioned in the database, or can refer to the singular number of the table's name (*acceptances* chapter).

The general form of the sentence is

find sentence1 (from|of sentence2) (where|that sentence3).

(from|of sentence2): This part is optional because the user is not obliged to specify the tables that are necessary in order to correctly compose an sql sentence. The system makes sure to fill this part, based on the elements mentioned in the sentence.

The parentheses state an optional part of the sentence and the sentence is composed only in this order.

The words *find*, *from*, *of*, *where*, *that*, *unique* relate to the sql keywords aforementioned. For all 3 sentences (**sentence1, sentence2, sentence3**), beside the elements that will be stated below, the user can use any English word that accommodates them in the composition of the sentence, provided that this word is not a system-bound expression (see *system-bound expressions* chapter).

Copy rights

Sentence 1:

In this part of the query the user must state at least one element of the database’s tables. For the extra elements that they want to state, they have to separate them with a comma or the word *and*. Finally, the user can also use the word *unique* that relates to the sql keyword *distinct*. In this part of the query the user refers to the elements that they wish to be returned by the query that will be done to the database. For example, an acceptable *sentence1* is:

find unique cname and cid

Sentence2:

In this part of the query the user must state the name of at least one table of the database. For extra tables that they want to state, they have to separate them with a comma or the word *and*. In this part of the query the user can optionally refer to tables, whose elements are stated in the query. For example, an acceptable *sentence2* is:

from tables agents, customers

Sentence3:

In this part of the query the user must state the preconditions that are required in order to receive the desired results from the query. This means that the user must state the preconditions that the results must keep. Amongst these preconditions the user can utilize one of the logic operands *and*, *or*. A precondition must be composed in the following form:

Element operand value.

Element: With the term element we refer to an element of a table of the database.

Operand: The operands that can be used are:

equal (=), not equal (!=), greater than/greater (>), less/less than (<), greater or equal than/greater or equal (>=), less or equal than/less or equal (<=) or none of the above. In the last case, our system uses the =.

Value: Can be an arithmetical value or an alphanumeric. If it is an alphanumeric, it must be written in (“ “)

Examples:

cid less than 12, aname = ”New York”, pid is 15

Moreover, in this part the user can use names of tables that assist them in optimal composition of the sentence. A possible reason to use table names in this part is to relate tables. The system handles this occasion as we explain in the paragraph (join). For example, an acceptable *sentence3* is:

that the customer places an order through the agent

Some examples of complete user sentences are the following:

Input:

1.find unique cname and aname that the customer places an order through the agent

2.find the pid of products that an order has been placed

3.find cid of customers that has cid greater than 12

4.find the aid and aname of agents that their acity is equal to "New York"

5.find the price of the products that the agent with aid 12 made an order on month "January".

6.find the acity that lives the agent with aname equal to "Smith".

Output:

1. select distinct customers.cname , agents.aname from customers , orders , agents where customers.cid = orders.cid and agents.aid = customers.aid and agents.aid = orders.aid
2. select products.pid from products , orders where products.pid = orders.pid
3. select customers.cid from customers where customers.cid > 12.00000
4. select agents.aid , agents.aname from agents where agents.acity = "New York"
5. select products.price from products , agents , orders where agents.aid = 12.000000 and orders.month = "January" and products.pid = orders.pid and agents.aid = orders.aid
6. select agents.acity from agents where agents.aname = "Smith"

Copy Rights

3.3 System-bound Expressions

The expressions that we have bounded in our system are the following:

A) The names of the tables and fields of each database scheme that the system runs.

B) The words that relate to sql keywords that we have mentioned above (from, find, of, that, where, unique).

C) The following associations and comparison operands: and, or, the comma punctuation mark (,), equal (=), not equal (!=), greater than/greater (>), less/less than (<), greater or equal than/greater or equal (>=), lesser or equal than, lesser or equal (<=) and the quotation marks (“ ”).

3.4 Final Processing and Result Extraction

During the syntax analysis of the user’s query, sql code is produced. When the analysis is complete, the produced code has been based on the composition method of the query, but it does not represent a sound sql sentence form. In the final stage we process the produced query, adding to it the parts that are required to render it a sound sql query. Specifically, we examine the elements that the query refers to and we fill in all the necessary tables in the *from* field of the sql query. Thereupon, based on the tables that the query refers to, we add at the *where* field the necessary equalities, taking into consideration the relation between the tables

4. CONCLUSION

Conclusively, the system that we have created offers the following capabilities:

a) It offers the ability to simple or unspecialized users to retrieve data from a database writing their queries in natural language, keeping only certain simple limitations.

b) The system works for every relational database scheme. This represents one of the primary advantages of this paper, compared to preceding papers that utilized a specific database.

c) We checked its operation with multiple samples of natural language queries, thus making certain that it works for the majority of them. As we have mentioned above, it works for simple occasions select from where. The only cases that it does not work are the queries that require more than one instances of the same table.

As we have stated above, our system works only for simple sql sentences. For the operation of our system we followed specific techniques that the system, based on them, can be developed to cover additional and more complex queries. The expandable parts of the system are stated below.

The system, as we have stated above, can work on any database. Exceptions are the databases with tables that have as keys more than one of their elements. Easily, with some changes in the system’s code, it is possible to support those databases.

The keywords that our system supports are the *select, distinct, from* and *where.* Following the created reasoning, it is possible to expand the system adding more sql keywords. This way, a much greater range of queries can be supported.

For the certain keywords that we support we offer a relation to words that can be used by the user for the query composition. The relation we offer is limited, but it is very easy to add more words or phrases in the sqlkeys.txt file. For example, by adding to the file the line *select : select*, the system will also support the word *select.*

Our system receives dynamically a database as input. This is a very important feature concerning the perspectives of our system. Below we analyze specific examples of our system’s usability.

One of its use perspectives is in an organization or company that uses a database. Giving as input to our system the database scheme of the corresponding company, it will be feasible for any employee that has knowledge over the usage of the system to retrieve information from the database, without them having any specialized knowledge over databases.

Another use example is in a course. If the course’s material is stored in a database, the students can easily, with the use of our system, retrieve information concerning the course from the database.

Copy Rights

REFERENCES

Book

Vlachavas I.et. Al, 2002, Artificial Intelligence, 2002, Gartagani Publishing.

Seymour Knowles 1991, A Natural Language Database Interface For SQL-Tutor, Honours Reports, University of Canterbury, 1999, Available from Internet: http://www.cosc.canterbury.ac.nz/research/reports/ HonsReps/1999/hons_9904.pdf

Androutsopoulos I. 1992, Interfacing a Natural Language Front-End to a Relational Database. MSc thesis, University of Edinburgh, 1992

E.F. Codd 1974, Seven Steps to RENDEZVOUS with the Casual User. In J. Kimbie and K. Koffeman, editors, Data Base Management. North-Holland Publishers, 1974

Grosz J. et. al. 1987, TEAM: An Experiment in the Design of Transportable Natural-Language Interfaces. Artificial Intelligence,32:173–243, 1987

Hendri G. et. al. 1978, Developing a Natural Language Interface to Complex Data. ACM Transactions on Database Systems, 3(2):105–147,1978.

Thompson B. and Thompson F. 1985, ASK is Transportable in Half a Dozen Ways ACM Transactions on Office Information Systems, 3(2):185–203, April 1985.

Rich E. and Knight K.1991,, Artificial Intelligence, Second Edition, 1991, McGrawttill.

Journal

Androutsopoulos I. et. al. 1995 , Natural Language Interfaces to Databases – An Introduction, Journal of Language Engineering, 1995. pages 28-81

Warren D. and Pereira F. 1982, An Efficient Easily Adaptable System for Interpreting Natural Language Queries. Computational Linguistics, 8(3-4):110–122, July-December 1982

Conference paper or contributed volume

Androutsopoulos I. et. al. An Efficient and Portable Natural Language Query Interface for Relational Databases. In P.W. Chung, G. Lovegrove, and M. Ali, editors, Proceedings of the 6th International Conference on Industrial & Engineering Applications of Artificial Intelligence and Expert Systems, Edinburgh, U.K., pages 327–330. Gordon and Breach Publishers Inc., Langhorne, PA, U.S.A., June 1993.ISBN 2–88124–604–4

T. Radhakrishnan, et. al. 1983, R. Castillo, Spoken Responses to Database Queries, proceedings of the international Conference of Systems, Man and Cybernetics, Dec 30 1983 - Jan 1984, India Vol. II, pages 825-829, 1983 IEEE Catalog No 83CH 1962-0, ISBN 0-08382683-0

Shan Wang et. al. 1999, Nchiql: a Chinese natural language query system to databases. 1999 International Symposium on Database Applications in Non-Traditional Environments (DANTE'99), pages 453-460.

Woods W. et. al. 1972, The Lunar Sciences Natural Language Information System: Final Report. BBN Report 2378, Bolt Beranek and Newman Inc.,Cambridge, Massachusetts, 1972.

Copy Rights

APPENDIX A: THE GRAMMAR OF THE NATURAL LANGUAGE OF OUR SYSTEM

```
S -> STM1 STM2 STM3
      | ε
EXPR1 -> select EXPR1
            |distinct COND1
            | COND1
COND1 -> field LIST1
LIST1 ->comma field LIST1
      | and field LIST1
      | ε
```

```
STM2 -> from COND2
        | ε
COND2 -> table LIST2
LIST2 -> comma table LIST2
        | and table LIST2
        | ε
STM3 -> where COND3
        | ε
COND3 -> CLAUSE LIST3
CLAUSE -> field OP VALUE
        | table
LIST3 -> and CLAUSE LIST3
        | or CLAUSE LIST3
        | CLAUSE LIST3
        | ε
OP -> equal
        | greater than
        | less than
        | greater or equal than
        | less or equal than
        | not equal
        | ε

VALUE -> string
        | number
```

Where S is the start symbol of the grammar.

Where ε is the blank word.

The non-terminal words are represented with uppercase letters, while the terminal words are represented with lowercase letters

Copy Rights

A MANET WITH CACHING CAPABILITIES VISUALIZATION TOOL

F.J. González-Cañete, L.B. Ríos-Sepúlveda, E. Casilari, and A. Triviño-Cabrera
Dpto. Tecnología Electrónica, University of Málaga
ETSI Telecomunicación, Campus de Teatinos, Universidad de Málaga, 29071 Málaga (Spain)

ABSTRACT

This paper presents a desktop application tool for visualizing simulations of Mobile Ad Hoc Networks with caching capabilities. This tool uses the log files generated by any network simulator that implements the required log format and the mobility file with the NS2 format in order to represent both, the network traffic and the mobility respectively. This tool works as an interface to represent the network activity and the node mobility by means of animations. On the other hand, this tool also calculates and shows, on real time and at any simulation time, the statistics needed to evaluate the MANET performance as well as the cache states. Finally, reports in PDF format including the statistics of each node and the global statistics can be generated. The application is implemented using the Java language and hence it is multiplatform.

KEYWORDS

MANET, caching, simulations, visualization tool.

Copy Rights

1. INTRODUCTION

A Mobile Ad Hoc Network (MANET) is a set of autonomous mobile terminals (MT) that can communicate among them using wireless links. As the MTs are forwarding packets to the MT in their coverage area, some kind of routing protocol is needed to make the routing decisions. The management of this kind of networks is performed in a decentralized way and all the nodes in the network have to cooperate forwarding and routing the packets to the other MTs in the network. The main characteristics of the MANETs are:

- Dynamic topology - Due to the mobility pattern, the MTs can enter or leave the coverage area of other MTs. This situation forces to recalculate the routes to other MTs in order to enable packet forwarding.
- Limited bandwidth and variable capabilities – The wireless medium is characterized by a reduced bandwidth and a greater error probability than the wired medium, since the radio medium is always shared and prone to interferences and packet collisions.
- Limited batteries and processing capabilities – Due to the mobile nature of this kind of networks, the mobile devices have to be portable (wearable in some cases) and hence the processing and battery capabilities are also restricted.

Due to the kind of mobile devices available in the MANET (such as laptops and smart cellular devices) more and more users demand the access to external networks, data servers or the Internet, thus these kind of networks have to be prepared to use these services. Figure 1 depicts an architecture where a MANET is connected to the Internet using two alternative Access Routers provided by two Access Networks.

Because of the nodes mobility the Access Routers can be out of the coverage area of any mobile node in the MANET. This causes the disconnection to the external networks. In order to avoid this situation some caching mechanisms have been proposed. According to these mechanisms nodes may store some of the data requested to the servers and cooperate to share information about where a valid copy of the data is stored in the MANET.

Developing, validating and evaluating cooperative caching architectures is hard work because of the distributed nature of the ad hoc networks. Moreover, the existing MANET simulators do not allow to visualise all the internal structures of the nodes or the network traffic as the simulation evolves. Due to this limitation we propose the implementation of a MANET visualization tool that allows to observe the nodes movements as well as the information related to the caching schemes implemented. On the other hand the network visualization tool transforms the difficulty to interpret trace files through a friendlier interface.

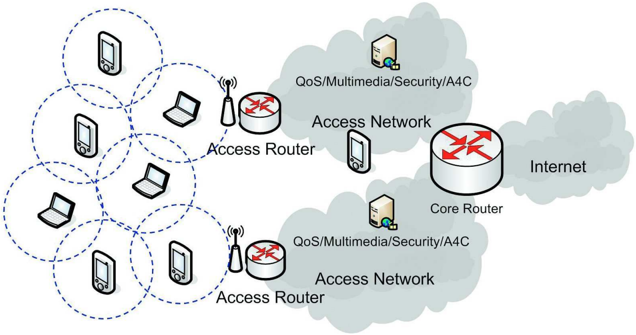


Figure 1. MANET with Internet connection

The rest of this paper is organized as follows. Section 2 describes some alternative visualization tools for MANETs pointing their advantages and flaws. Section 3 comments the caching architecture supported by the implemented application. Section 4 illustrates the application capabilities and specifications. Finally, Section 5 outlines the main conclusion of this work and proposes future works.

2. RELATED WORK

The NAM tool (Network AniMator) (NAM, 2010) is the default NS2 (Network Visualization) (NS2, 2010) visualization tool and it is able to process a great amount of data and to calculate some statistics about the simulations. NAM was designed to provide a graphical user interface in order to configure simulations of wired network topologies, allowing to show the links created and the packets flows. Unfortunately, it has not been adapted to the mobile nature of the MANETs. There was a project called ad-hockey (ad-hockey, 2010) created by the Monarch project (Monarch, 2010) in the Carnegie Mellon University started in the late 90s in order to add this capability, but it was discontinued.

iNSpect (interactive NS-2 protocol and environment confirmation tool) (Kurkowski, 2005) is a visualization tool developed using C++ that allows the analysis of wireless networks simulated by NS2. Its main characteristic is that it is able to manage different input log files, accepting the log and mobility files generated by NS2, but it also needs a special kind of trace file denoted by vizTrace. In addition, iNSpect can also validate the NS2 because it can test if NS2 correctly manages the mobility pattern. Finally, this tool also calculates statistics parameters about the simulated networks. However iNSpect does not offer information about the current simulation such as the number of processed packets, size, source and destination IP addresses of each hop and does not permit the examination of the internal structures of the mobile nodes.

Huginn (Scheuermann, 2005) is another visualization tool for wireless networks using NS2 implemented using C++. Its main capability is the usage of 3D graphics for representing the networks and a sophisticated schema to display the information. The user can specify a flow diagram that defines the data filtering. Thus, the user can specify the kind of data in which the application has to concentrate on. However Huginn has the same flaws as the previous tools, as it does not represent the internal structures implemented in the mobile nodes.

EXAMS (EXtensible Animator for Mobile Simulations) (Livathinos, 2009) is a full simulator written in Java that adds functionalities which are not available in the above mentioned tools, such as the possibility to represent the nodes internal state, to view the coverage area of the nodes and even calculate statistic data about the network performance. Unfortunately it does not implement any kind of caching mechanisms, which is the main goal of our studies. In addition EXAMS does not perform a node position interpolation so it does not perform a validation of the nodes movement.

Finally, MobiSim (Mousavi, 2007) is a free application developed with the aim to support the investigations in MANETs and the study of the mobility in these kind of networks. The user can create mobility scenarios and simulate them graphically. It supports a wide range of mobility patterns that can also

Copyrights

be merged. However MobiSim is only able to manage mobility models and not the traffic along the network. It is only a mobility pattern generator and a visualization tool for the movements.

Additionally, we must consider that all the above mentioned visualization tools, except EXAMS, are only Linux compatible and hence, they are not possible to be used in other operating systems such as Microsoft Windows or MacOS.

3. CACHING ARCHITECTURE

In this section we present the caching architecture for ad hoc networks supported by the visualization tool. In this scheme the wireless nodes in the network request documents that are located in data servers. The procedure works as follows: a node requests a document to a data server; the request is routed through the ad hoc network using the routing protocol defined for this network. When the data server receives this request it responds by sending the document to the requester node. This scheme is similar to other request-response algorithms proposed in the literature (Yin, 2006).

Each mobile node in the wireless network will implement a local cache where the requested documents will be stored when they are received. In that way if the mobile node needs the same document in the near future it will be served by the local cache instead of the remote server. This situation is called a local cache hit. There are some local cache parameters that have to be taken into consideration:

- The storage space reserved to store the documents will determine the amount of documents that will fit in the local cache. As the cache size increases more documents will be stored and the probability of a local cache hit will also increase.
- Each document is associated to a TTL (Time To Live). This parameter defines when the document will expire and hence it has to be deleted from the local cache because the information is obsolete.
- The replacement policy defines the algorithm that decides which documents stored in the local cache will be evicted in order to make room for the new ones. Many replacement policies have been proposed in the literature and they all try to select for eviction the documents with the lowest probability of being requested again in the near future.

In our caching architecture we proposed to use the LRU (Least Recently Used) replacement policy although the visualization tool supports any replacement policy.

On the other hand, the mobile nodes in the wireless network also work as a proxy for the other mobile nodes. Each node in the path of a request from the node requester to the data server will check if it has a valid copy of the requested document in its local cache. If so, the intermediate mobile node will directly reply to the requester node with the document. This situation is called an interception cache hit and it reduces the response time and the server load as the requests do not reach them.

Since each mobile node in the wireless network has to forward the requests and responses acting as a proxy, they can also store information about the position where documents are stored in the other mobile nodes. In this way, each time a node receives a request to forward it stores that the document requested may be stored in the local cache of the requester node and how far the node is (in number of hops). Moreover, when a response is forwarded each node will also annotate which node replied and the corresponding distance in hops. The TTL of the document is also taken into account in order to set a validation time for this information. Each time a node receives a request to forward it searches for a mobile node that has a copy of the requested document in its local cache and that it is closer to the origin node than the data server. If so, the request is redirected to the node that is supposed to have the document, which will reply to the requester node with the document. This situation is called a redirection cache hit.

Unfortunately, the redirection of requests has some drawbacks that have to be considered:

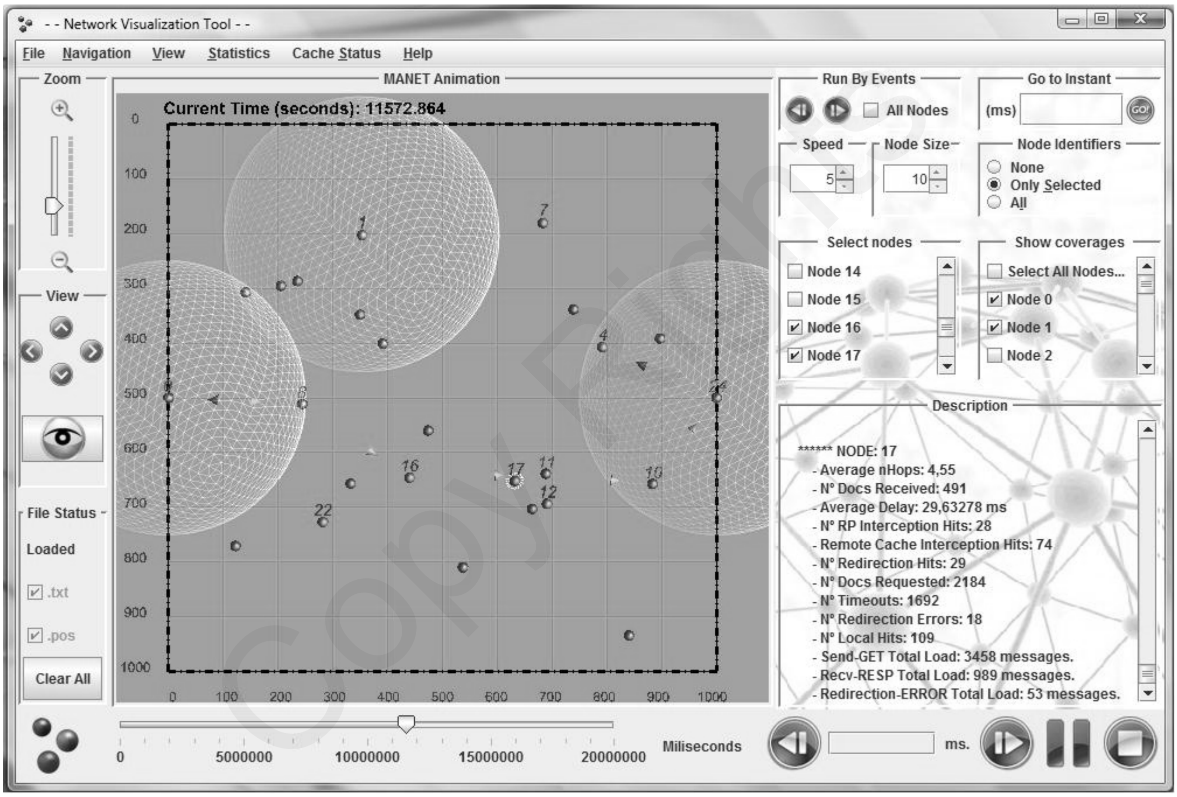
- The nodes mobility and disconnections could cause the information stored about the documents location in the mobile network and the distances among the nodes to be invalid.
- The replacement policies can also decide to evict certain documents of a local cache and hence, the information about these documents in this node will also be invalid.

The previous situations could cause that a node decides to redirect a request to another node which is not reachable and hence, the request will never reach its destination. This can be partially solved implementing a timeout in the soliciting node in order to request the document again if this timeout is reached. When a mobile node receives a redirected request and it does not have a valid copy of the document (due to the replacement policy), the mobile node will send a redirection error message to the node that decided the redirection in order to update the information about the location of this document.

The preceding caching architecture was previously presented and evaluated by the authors in (González-Cañete et al, 2009a) and (González-Cañete et al, 2009b). This caching policy is shown to reduce the network traffic and server load. In addition, the energy consumption of each node is also decreased because the amount of messages to be forwarded is also diminished.

4. THE VISUALIZATION TOOL

The visualization tool proposed has been developed using the Java language and the Java3D API. As the Java language is multiplatform, the application can be executed in any platform that supports it. Nowadays these platforms are Microsoft Windows, Linux, Solaris and Mac OS X although the tool has only been tested in the Microsoft Windows and Linux platforms. The application main window can be observed in Figure 2.



Current Time (seconds): 11572.864

□ All Nodes

Speed 5

Node Size 10

○ None
● Only Selected
○ All

□ Node 14
□ Node 15
☑ Node 16
☑ Node 17

□ Select All Nodes...
☑ Node 0
☑ Node 1
□ Node 2

Loaded
☑ .txt
☑ .pos

```
****** NODE: 17
  - Average nHops: 4,55
  - N° Docs Received: 491
  - Average Delay: 29,63278 ms
  - N° RP Interception Hits: 28
  - Remote Cache Interception Hits: 74
  - N° Redirection Hits: 29
  - N° Docs Requested: 2184
  - N° Timeouts: 1692
  - N° Redirection Errors: 18
  - N° Local Hits: 109
  - Send-GET Total Load: 3458 messages.
  - Recv-RESP Total Load: 989 messages.
  - Redirection-ERROR Total Load: 53 messages.
```

Figure 2. Visualization tool main window

The visualization tool accepts two input files:

- The mobility file – It includes information about the nodes positions, movements and speeds as well as the simulation area dimensions. The mobility file must follow the NS2 mobility file format and it can be created with any compatible generator such as the BonnMotion scenario generation tool (Aschenbruck, 2010)
- The traffic file – It specifies the information about the traffic along the network and the local and redirection caches states for each node. The traffic file format specifications are available at http://pc23te.dte.uma.es/FilesFormat.pdf.

When the mobility file is loaded the scenario is depicted in the MANET Animation panel of the application. The simulation area and its dimensions are shown using a grid of 100 meters. The mobile nodes are represented as spheres that can be selected in order to view their identification number and coverage area. The tool can be used to validate the generated scenarios because it is able to visualize and animate the nodes positions along the network according to the mobility file without specifying any traffic file. On the other hand, when the traffic file is loaded in conjunction with the mobility file, the application is ready to visualize the traffic and node movement in the network.

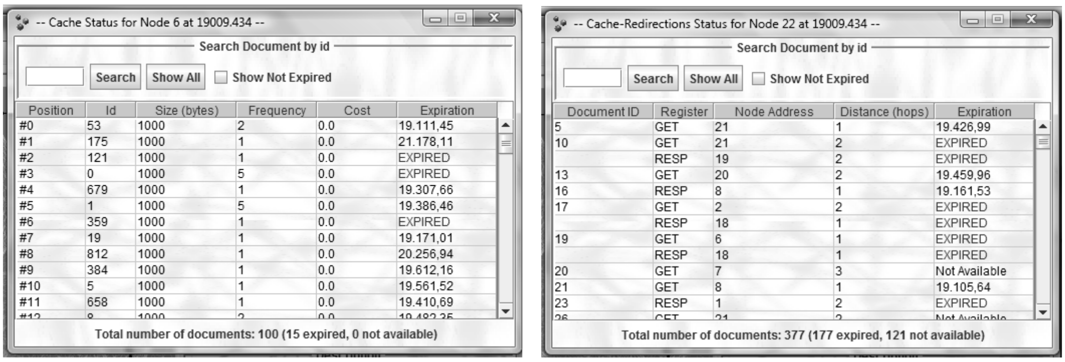
The application capabilities will be enumerated as the panel's functionalities are commented:

- Zoom and View panels – These panels allow to change the point of view of the camera that represents the scenario. The camera can be moved in the four basic directions as well as to zoom in and out in the scenario.
- Playback panel – This panel contains the play back functionalities. With the slider control the simulation can be positioned instantly at any simulation time. On the other hand, the visualization can be played, paused and stopped. Finally the visualization can run at fixed and controlled time steps defined by the user.
- Run By Events panel – This panel permits to run the visualization and stop automatically once an event has occurred at any of the selected nodes. The considered events are: the node sends, forwards or receives a message, the node reaches its destination coordinates in the mobility model and/or the local or redirection caches are modified.
- Go To Instant panel – An absolute simulation time can be specified in order to set the visualization instant to this time.
- Speed panel – By default the simulation speed is at real time. This control allows to speed up or down the visualization.
- Node Size panel – The nodes sizes can be increased or decreased using this control.
- Node Identifiers panel – The node identification number can be viewed for all, none or only the selected nodes in the MANET.
- Select Nodes panel – The nodes can be selected or deselected checking the node identifiers shown in a list.
- Show Coverage panel – Similarly, the node coverage area visualization can be enabled or disabled selecting the nodes in the corresponding node list.
- Description panel – This panel lists and depicts the events and main statistics of the selected nodes.
- MANET Animation panel – This is the main panel in the application and represents the scenario to be simulated. It represents the mobile nodes and their movements along the simulation area. Moreover, the mobile nodes can be selected directly clicking over the sphere with the left mouse button. In addition, if the right mouse button if pressed over a node a menu will appear offering to show the statistics, coordinates, identifier, coverage area and the local and redirection cache of the node.

At any time in the simulation the local and redirection caches of the nodes can be shown. Figure 3.a illustrates the local cache content window. In this popup window the document identification, size, number of accesses, cost (for cost based replacement policies) and expiration time are listed ordered by the positions in the local cache. The document located in the highest position in the local caches will be the next document to be evicted. This local cache window also allows to filter the documents by the document identification number or showing only the documents which have not expired. Figure 3.b depicts the redirection cache window where the document identification number (*id*), the type of information register (could be GET or RESP depending on whether the information was obtained by a request or a response respectively), the node identification where the document is located, the distance in hops to the node that contains the document and the information expiration time are shown. Similarly to the local cache window the information listed in the redirection cache window can also be filtered showing only the information which is not obsolete as well as selecting a specific document.

Aiming at obtaining a better comprehension of the events that occur in the wireless network the visualization tool also includes some visual indicators that facilitate the discrimination of certain situations. In that way, when a mobile node is trying to send a request a halo is displayed informing about this fact (Figure 4.a). Similarly, the traffic between mobile nodes is represented using animated arrows indicating the traffic direction (Figure 4.b).

Copy Rights



-- Cache Status for Node 6 at 19009.434 --

Search Document by id

Position	Id	Size (bytes)	Frequency	Cost	Expiration
#0	53	1000	2	0.0	19.111,45
#1	175	1000	1	0.0	21.178,11
#2	121	1000	1	0.0	EXPIRED
#3	0	1000	5	0.0	EXPIRED
#4	679	1000	1	0.0	19.307,66
#5	1	1000	5	0.0	19.386,46
#6	359	1000	1	0.0	EXPIRED
#7	19	1000	1	0.0	19.171,01
#8	812	1000	1	0.0	20.256,94
#9	384	1000	1	0.0	19.612,16
#10	5	1000	1	0.0	19.561,52
#11	658	1000	1	0.0	19.410,69
#12	8	1000	2	0.0	19.482,35

Total number of documents: 100 (15 expired, 0 not available)

(a)

-- Cache-Redirections Status for Node 22 at 19009.434 --

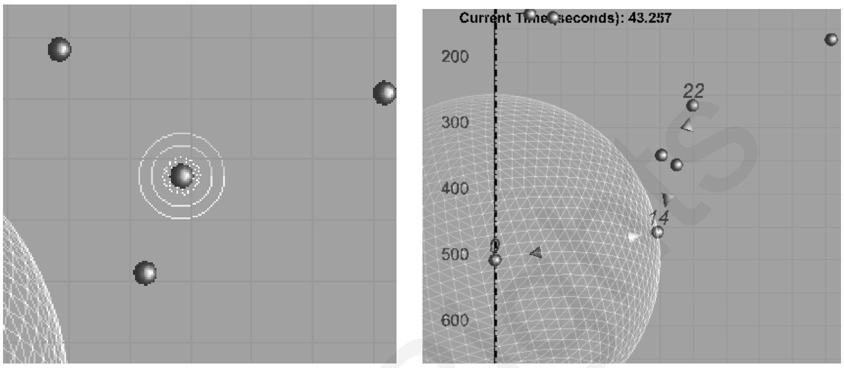
Search Document by id

Document ID	Register	Node Address	Distance (hops)	Expiration
5	GET	21	1	19.426,99
10	GET	21	2	EXPIRED
	RESP	19	2	EXPIRED
13	GET	20	2	19.459,96
16	RESP	8	1	19.161,53
17	GET	2	2	EXPIRED
	RESP	18	1	EXPIRED
19	GET	6	1	EXPIRED
	RESP	18	1	EXPIRED
20	GET	7	3	Not Available
21	GET	8	1	19.105,64
23	RESP	1	2	EXPIRED
26	GET	21	2	Not Available

Total number of documents: 377 (177 expired, 121 not available)

(b)

Figure 3. Local cache content window (a) and remote cache window (b)



(a) (b)

Figure 4. Visual helpers. Halo informing about a request (a) and traffic (b)

Apart from the above mentioned visualization capabilities the application also calculates performance and statistics parameters about the simulation at any simulation time. These parameters could be used as an example to compare the performance among network configurations or cache sizes. The main statistics that the application calculates for each node are:

- Number of sent requests and received documents.
- Number of timeouts – When a mobile node requests a document it waits for a certain amount of time to receive the response. If the document does not reach the originator node during this time, the request is considered lost and the document is requested again.
- Average delay and hops – For each document received the time the request took to be served is measured as well as the number of network hops needed to transfer the requests and the reply.
- Number of cache hits and errors – The local cache hits, interception hits, redirection hits and redirection errors are stored.
- Traffic – The application calculates the amount of requests, responses and errors sent or forwarded by each node.

Additionally, the statistics can be exported to a PDF file where the previous values are saved for each node at the current simulation time. Histograms about each previous parameter and all nodes are also included in the report as well as the average values for the entire network.

5. PERFORMANCE EVALUATION

In order to evaluate the application limits, 27 mobile scenarios have been tested varying the number of mobile nodes (25, 50 and 100), the node speed (1, 3 and 5 m/s) and the mean time between requests (5, 25, and 50 seconds). The simulation time was set to 20000 seconds that is twice the greater simulation time found in the caching in ad hoc networks literature. These tests were performed using an Intel Core 2 Duo 2.0

GHz computer using the Ubuntu 9.10 (32-bits) operating system and the maximum memory allocation value allowed for the Java Virtual Machine (JVM) in the operating system selected (2.5 GB). When the application failed because of memory limitations the simulation time was reduced until the visualization was possible.

Table 1 summarizes the scenarios tested as well as the trace file size, the number of events and the memory consumed by the JVM.

Table 1. Application performance evaluation scenarios

Nodes	Speed (m/s)	Mean time between requests (s)	Trace file size (MB)	Events (x 10^3)	Simulation time (s)	Memory load (MB)
25	1	5	82	1421	20000	970
		25	36	629	20000	550
		50	24	424	20000	510
	3	5	92	1590	20000	960
		25	36	637	20000	730
		50	22	395	20000	510
	5	5	92	1590	20000	1060
		25	37	644	20000	650
		50	23	405	20000	510
50	1	5	231	3862	20000	1640
		25	60	1016	20000	700
		50	33	549	20000	550
	3	5	244	4120	20000	1750
		25	66	1115	20000	730
		50	35	592	20000	560
	5	5	244	4159	20000	1760
		25	69	1167	20000	660
		50	38	638	20000	620
100	1	5	322	5369	13500	2540
		25	125	2091	20000	1050
		50	66	1088	20000	820
	3	5	344	5791	13500	2560
		25	140	2360	20000	1140
		50	73	1225	20000	930
	5	5	353	5982	13500	2560
		25	153	2594	20000	1450
		50	76	1273	20000	840

Copy Rights

As can be observed, as the number of nodes or the nodes' speed increases or the mean time between requests decreases, the number of events in the network and the trace file size are also increased. For all the scenarios except the ones with 100 nodes and 5 requests per second the visualization was performed without problems. For these very loaded scenarios the simulation time had to be reduced until 13500 seconds that is a reasonable and useful simulation time.

6. CONCLUSION

In this work a multiplatform visualization tool for mobile ad hoc networks with caching capabilities has been presented. The network behaviour accepted is a request/response protocol where the mobile nodes request documents to data servers that reply with the documents. Each mobile node in the network implements a local cache and can perform as a proxy for the other mobile nodes because each node is able to reply to the requests when it has a valid copy of the requested document in its local cache. In addition, the mobile nodes learn, using the traffic they forward, where and how far the documents are located in the network and hence they can decide to redirect a request to a nearer node than the original requested destination.

The implemented tool allows to visualize and animate the wireless nodes moving along the network. Moreover, if the traffic trace is defined the application also represents the requests and responses using animations. The local and redirection cache can be visualized for each node and even the information presented can be filtered in order to focus on the more relevant cache entries for the simulation.

The application offers several simulation playback controls allowing to view the simulation in real time, to jump to an absolute simulation time, to step back and forward a certain amount of time and to play event by event. The event by event playback functionality is very useful to monitor the node activity.

Additionally, the application calculates the most important statistical parameters that make it possible to study the performance of the wireless network and the caching mechanism. A detailed PDF report is also generated including all the statistics calculated for each node, the histograms for each statistical parameter as well as the global network statistics.

The performance of the application has been widely evaluated using several MANET scenarios and we concluded that the main limitation is the amount of memory allocated for the Java Virtual Machine.

Finally, we must emphasize that the main target of the presented tool is to study, develop and debug cooperative caching architectures in mobile ad hoc networks. This is the unique network visualization tool that allows having access to the internal caching structures of each node in the network apart from being a multiplatform application as it has been developed using the Java technologies.

As a future work we propose to enhance the functionalities offered by the application such as supporting broadcast messages that will enable to support broadcast caching architectures.

Copy Rights

ACKNOWLEDGEMENT

We would like to thank Adela Isabel Fernández Anta for revising the syntax and grammar of this paper. This work was partially supported by the public project TEC2009-13763-C02-01.

REFERENCES

ad-hockey home page, 2010, http://www.monarch.cs.rice.edu/cmu-ns.html

Aschenbruck,N., et al., 2010. BonnMotion – A Mobility Scenario Generation and Analysis Tool. *Proceedings of the 3rd International Conference on Simulation Tools and Techniques (SIMUTOOLS 2010)*. Malaga, Spain.

González-Cañete, F.J., et al., 2009a. Proposal and Evaluation of an Application Level Caching Scheme for Ad Hoc Networks. *Proccedings of the 5th International Wireless Communications and Mobile Computing Conference (IWCMC 2009)*. Leipzig, Germany, pp. 952-957.

González-Cañete, F.J., et al., 2009b. Proposal and evaluation of a Caching Scheme for Ad Hoc Networks. *Proceedings of the 8th International Conference on Ad Hoc Networks and Wireless (Ad-Hoc Now 2009)*. Murcia, Spain, pp. 366-372

Kurkowski, S., et al.,2005. A Visualization and Animation Tool for NS-2 Wireless Simulations: iNSpect. *Proceedings of the 13º IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*. Atlanta, USA, pp. 503-506.

Monarch project home page, 2010, http://www.cs.cmu.edu/afs/cs.cmu.edu/Web/People/dbj/monarch.html

Mousavi, S.M., et al., 2007. MobiSim : A Framework for Simulation of Mobility Models in Mobile Ad-Hoc Networks. *Proceedings of the 3rd IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (IEEE WiMob 2007)*. New York, USA,

NAM home page, 2010, http://www.isi.edu/nsnam/nam/

Livathinos, S.N., 2009. EXtensible animator for Mobile Simulations: EXAMNS, *Proceedings of the 17th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*. London, United Kingdom.

NS2 home page, 2010, http://www.isi.edu/nsnam/ns/

Scheuermann, B., et al., 2005. Huginn: A 3D Visualizer for Wireless ns2 Traces. *Proceedings of the 8º ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems*. Montreal, Canada, pp. 143-150.

Yin, L., and Cao, G., 2006. Supporting Cooperative Caching in Ad Hoc Networks. *In IEEE Transaction on Mobile Computing*, Vol. 5, No. 1, pp.77- 89.

SEQUENTIAL AND DISTRIBUTED HYBRID GA-SA ALGORITHMS FOR ENERGY OPTIMIZATION IN EMBEDDED SYSTEMS

Maha Idrissi Aouad[1], Lhassane Idoumghar[2, 1], René Schott[3] and Olivier Zendra[1]
[1] *INRIA Nancy - Grand Est / LORIA. 615, rue du Jardin Botanique, 54600 Villers-Lès-Nancy, France*
[2] *LMIA - MAGE, Université de Haute-Alsace. 4, rue des Frères Lumière, 68093 Mulhouse, France*
[3] *IECN - LORIA, Nancy-Université, Université Henri Poincaré. 54506 Vandoeuvre-Lès-Nancy, France*

ABSTRACT

Reducing memory energy consumption in embedded systems is crucial. In this paper, we propose new hybrid sequential and distributed algorithms based on Simulated Annealing (*SA*) and Genetic Algorithms (*GA*) in order to reduce memory energy consumption in embedded systems. Our algorithms outperform the Tabu Search (*TS*) approach. In fact, our hybrid algorithms manage to consume nearly from 76% up to 98% less memory energy than TS. Execution time savings for the distributed version (nearly from 72% up to 74% for a cluster of 4 PCs) are also recorded.

KEYWORDS

Distributed systems, embedded systems, genetic algorithms, memory management, optimization, simulated annealing.

1. INTRODUCTION

Embedded systems have become more energy greedy due to technology evolution. Indeed, they must integrate multiple complex functionalities which needs bigger battery and memory. Accordingly, memory will become the major energy consumer in an embedded system (ITRS, 2007). Hence, reducing this memory energy is crucial. In this paper, we will focus on software optimizations working on the memory management.

In order to save energy, most authors rely on Scratch-Pad Memories (*SPMs*) rather than cache memories and various related research directions have been investigated (Idrissi Aouad and Zendra, 2007). Although caches, help a lot with program speed, they are not the most appropriate for embedded systems. In fact, caches increase the system size and its energy cost because of cache area plus managing logic. Like cache, SPM consists of small, fast SRAM, but the main difference is that SPM is directly and explicitly managed at the software level, either by the developer or by the compiler, whereas cache requires extra dedicated circuits. Its software management makes it more predictable as we avoid cache miss cases. Compared to cache, SPM thus has several advantages (Idrissi Aouad and Zendra, 2007). SPM requires up to 40% less energy and 34% less area than cache (Ben Fradj et al., 2005). Additionally, manufacturing SPM cost is lower and a large variety of chips with SPM is available in the market (Adiletta et al., 2002). We will therefore use an SPM in our memory architecture.

The rest of the paper is organized as follows. Section 2 presents related work. Section 3 gives the memory energy model we used. Section 4 describes our optimization problem. Section 5 gives some preliminaries on GA and SA algorithms. Section 6 presents our hybrid GA-SA algorithms. Section 7 shows the experimental results. Finally, Section 8 concludes and gives some perspectives.

2. RELATED WORKS AND EXISTING HEURISTICS

Due to the reduced SPM size, authors try to optimally allocate data in it in order to realize energy savings. An approach is to place interesting data (the most frequently accessed (Wehmeyer et al., 2004), the most cache-

Copy Rights

conflict prone (Panda et al., 1997), etc.) in SPM (a low access cost memory) whereas the other data will be placed in main memory (*DRAM*) (a low storage cost memory). In order to determine interesting data, authors use *data profiling* to gather memory access frequency information. This information can be collected either statistically by analyzing the source code or dynamically by profiling the application (data size, access frequency, etc.). Thus, most effective strategies are:

Allocate data into SPM by access number/size (*BEH*): data are sorted according to their ratio (access number/size). The data with the highest ratio is allocated first into SPM as there is space available else it is allocated in DRAM. This heuristic uses a sorting method which can be computationally expensive for a large amount of data. Additionally, BEH will not work very well in a dynamic perspective where the SPM maximum capacity is not known in advance.

Allocate data into SPM based on Tabu Search (TS): details about how TS is implemented can be found in (Idrissi Aouad et al., 2010). TS is as energy efficient as BEH as shown in (Idrissi Aouad et al., 2010). In contrast, TS is easy to implement and since no sorting is necessary, unlike BEH, the corresponding time is saved. In addition to that, in a dynamic perspective where the SPM maximum capacity is not known in advance, TS will perform better than BEH.

3. MEMORY ENERGY ESTIMATION MODEL

In order to compute the energy cost of the system, we propose in this section an energy consumption estimation model for our considered memory architecture composed by an SPM, an instruction cache and a DRAM. Equation 1 gives the energy model where the three terms refer to the total energy consumed respectively in SPM, in instruction cache and in DRAM.

$$E = E_{tspm} + E_{tic} + E_{tdram} \qquad (1)$$

In this model, we distinguish between the two cache write policies: Write-Through (*WT*) and Write-Back (*WB*). In a WT cache, every write to the cache causes a synchronous write to DRAM. Alternatively, in a WB cache, writes are not immediately mirrored to DRAM. Instead, the cache tracks which of its locations have been written over and then, it marks these locations as dirty. The data in these locations is written back to DRAM when those data are evicted from the cache (Tanenbaum, 2005). Our aim is to minimize the detailed energy estimation model presented below:

$$
\begin{aligned}
E = {} & N_{spmr} * E_{spmr} && (2)\\
& + N_{spmw} * E_{spmw} && (3)\\
& + \sum_{k=1}^{N_{icr}} \left[h_{i_k} * E_{icr} + \left(1 - h_{i_k}\right) * \left[E_{dramr} + E_{icw} + (1 - WP_i) * DB_{i_k} * (E_{icr} + E_{dramw}) \right] \right] && (4)\\
& + \sum_{k=1}^{N_{icw}} \left[WP_i * E_{dramw} + h_{i_k} * E_{icw} + (1 - WP_i) * \left(1 - h_{i_k}\right) * \left[E_{icw} + DB_{i_k} * (E_{icr} + E_{dramw}) \right] \right] && (5)\\
& + N_{dramr} * E_{dramr} && (6)\\
& + N_{dramw} * E_{dramw} && (7)
\end{aligned}
$$

Lines (2) and (3) represent respectively the total energy consumed during a reading and during a writing from/into SPM. Lines (4) and (5) represent respectively the total energy consumed during a reading and during a writing from/into instruction cache. When, lines (6) and (7) represent respectively the total energy consumed during a reading and during a writing from/into DRAM. The various terms used in this energy model are explained in Table 1.

Copy Rights

Table 1. List of terms.

Term	Meaning
E_{spmr}	Energy consumed during a reading from SPM.
E_{spmw}	Energy consumed during a writing into SPM.
N_{spmr}	Reading access number to SPM.
N_{spmw}	Writing access number to SPM.
E_{icr}	Energy consumed during a reading from instruction cache.
E_{icw}	Energy consumed during a writing into instruction cache.
N_{icr}	Reading access number to instruction cache.
N_{icw}	Writing access number to instruction cache.
E_{dramr}	Energy consumed during a reading from DRAM.
E_{dramw}	Energy consumed during a writing into DRAM.
N_{dramr}	Reading access number to DRAM.
N_{dramw}	Writing access number to DRAM.
WP_i	The considered cache write policy: WT or WB. In case of WT, $WP_i = 1$ else in case of WB then $WP_i = 0$.
DB_{i_k}	Dirty Bit used in case of WB to indicate during the access k if the instruction cache line has been modified before $(DB_i = 1)$ or not $(DB_i = 0)$.
h_{i_k}	Type of the access k to the instruction cache. In case of cache hit, $h_{i_k} = 1$. In case of cache miss, $h_{i_k} = 0$.

4. OPTIMIZATION PROBLEM

Our problem is a combinatorial optimization problem. It is a kind of knapsack problem (Kellerer et al., 2004). We want to fill SPM that can hold a maximum capacity of C with some combination of data from a list of N possible data each with $size_i$ and $access\ number_i$ so that the access number of the data allocated into SPM is maximized. This problem has a single linear constraint, a linear objective function which sums the sizes of the data allocated into SPM, and the added restriction that each data will be in the SPM or not. If N is the total number of data, then a solution is just a finite sequence s of N terms such that $s[n]$ is either 0 or the size of the n_{th} data. $s[n] = 0$ if and only if the n_{th} data is not selected in the solution. This solution must satisfy the constraint of not exceeding the maximum SPM capacity $(i.e.\ \sum_{i=1}^{N} s[i] \leq C)$.

5. PRELIMINARIES ON GA AND SA ALGORITHMS

5.1 GA Algorithms

GAs (Sivanandam and Deepa, 2007) are adaptive methods which may be used to solve search and optimization problems. They are based on the genetic processes of biological organisms. By starting with a population of possible solutions and changing them during several iterations, GAs hope to converge to the fittest solution. Each solution is represented through a chromosome, which is just an abstract representation. The process begins with a set of potential solutions or chromosomes that are randomly generated or selected. Over many generations, natural populations evolve according to the principles of natural selection and survival of the fittest. For generating new chromosomes, GA can use both crossover and mutation techniques. Crossover involves splitting two chromosomes and then, for example, combining one half of each chromosome with the other pair. Mutation involves flipping a single bit of a chromosome. The chromosomes are then evaluated using a certain fitness criterion and the ones which satisfy the most this criterion are kept while the others are discarded. This process repeats until the population converges toward the optimal solution. There are several advantages to GA such as their parallelism and their liability. They require no knowledge or gradient information about the response surface, they are resistant to becoming trapped in local optima and they perform very well for large-scale optimization problems. GAs have been used as heuristics to solve difficult problems (such as NP-hard problems) for machine learning and also for evolving simple

Copyrights

programs. Applications of Genetic Algorithms include: nonlinear programming, stochastic programming, signal processing and combinatorial optimization problems.

5.2 SA Algorithms

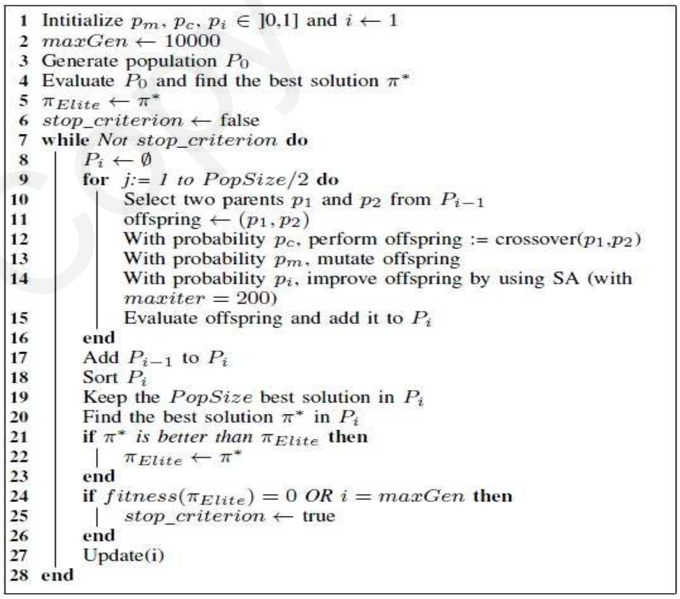
SA (Kirkpatrick et al., 1983) is a probabilistic variant of the local search method, but it can, in contrast, escape from local optima. SA is based on an analogy taken from thermodynamics: to grow a crystal, we start by heating a row of materials to a molten state. Then, we reduce the temperature of this crystal melt gradually, until the crystal structure is frozen in. A standard SA procedure begins by generating an initial solution randomly. At initial stages, a small random change is made in the current solution s_c. Then the objective function value of the new solution s_n is calculated and compared with that of the current solution. A move is made to the new solution if it has better value or if the probability function implemented in SA has a higher value than a randomly generated number. Otherwise a new solution is generated and evaluated. The probability of accepting a new solution is given as follows:

$$p = \begin{cases} 1 & if \quad f(s_n) - f(s_c) < 0 \\ exp\left(\frac{-|f(s_n)-f(s_c)|}{T}\right) & otherwise \end{cases} \tag{8}$$

The calculation of this probability relies on a parameter T, which is referred to as temperature, since it plays a similar role as the temperature in the physical annealing process. To avoid getting trapped into a local minimum point, the rate of reduction should be slow. In our problem the following method to reduce the temperature has been used, where $i = 0, 1, \ldots$ and $\gamma = 0.09$.

$$T_{i+1} = \gamma\, T_i \tag{9}$$

Thus, at the start of SA most worsening moves may be accepted, but at the end only improving ones are likely to be allowed. This can help the procedure jump out of a local minimum. The algorithm may be terminated after a certain volume fraction of the structure has been reached or after a pre-specified runtime.



```
Intitialize $p_m, p_c, p_i \in ]0,1]$ and $i \leftarrow 1$
$maxGen \leftarrow 10000$
Generate population $P_0$
Evaluate $P_0$ and find the best solution $\pi^*$
$\pi_{Elite} \leftarrow \pi^*$
$stop\_criterion \leftarrow$ false
while Not $stop\_criterion$ do
    $P_i \leftarrow \emptyset$
    for $j$:= 1 to $PopSize/2$ do
        Select two parents $p_1$ and $p_2$ from $P_{i-1}$
        offspring $\leftarrow (p_1, p_2)$
        With probability $p_c$, perform offspring := crossover($p_1$,$p_2$)
        With probability $p_m$, mutate offspring
        With probability $p_i$, improve offspring by using SA (with
        $maxiter = 200$)
        Evaluate offspring and add it to $P_i$
    end
    Add $P_{i-1}$ to $P_i$
    Sort $P_i$
    Keep the $PopSize$ best solution in $P_i$
    Find the best solution $\pi^*$ in $P_i$
    if $\pi^*$ is better than $\pi_{Elite}$ then
        $\pi_{Elite} \leftarrow \pi^*$
    end
    if $fitness(\pi_{Elite}) = 0$ OR $i = maxGen$ then
        $stop\_criterion \leftarrow$ true
    end
    Update(i)
end
```

Algorithm 1. Sequential hybrid GA-SA algorithm.

Copy Rights

6. OUR HYBRID GA-SA ALGORITHMS

6.1 Our Sequential Hybrid GA-SA Algorithm

The principles of our Sequential hybrid GA-SA (*GASA_Seq*) algorithm are described in Algorithm 1. In that algorithm GA uses SA as a third operator (called *Improve operator*). Other parameters are defined as below:

- **Initial subpopulation**: initial population P_0 is created randomly. In Algorithm 1 (and also in Algorithm 2), $PopSize = 30$ is the size of every population P_i. During each of the $maxGen$ generations, $PopSize$ offsprings are generated through the crossover of parents selected from the subpopulation.

$$Fitness(solution) = Total\_Number\_Access\_all\_data - Number\_Access(solution) \qquad (10)$$

- **Fitness evaluation**: the fitness function (see Equation 10) in hybrid GA-SA algorithms is typically the objective function we want to minimize in the problem. It serves for each individual to be tested for suitability to the environment under consideration.
- **Selection operator**: in our approach, we use a random selection that is a simple method for implementing fitness-proportionate selection. On each stage, an individual is randomly selected to be in the pool of parents for the next generation.
- **Crossover operator**: the crossover is a random process defined by a probability $p_c = 0.6$ and applied sequentially to pairs of parents chosen randomly in the population. It consists in exchanging parts of the genetic material of the parents in order to create two childes (offspring). In our approach we have used the two points crossover where two points are chosen randomly and the contents between these points are exchanged between two mated parents.
- **Mutation operator**: mutation operator changes the new offspring by flipping bits from 1 to 0 or from 0 to 1 (while solution remains feasible). Mutation operator can occur at each bit position in the array with some probability (in our algorithm $p_m = 0.06$).
- **Improve operator**: with probability $p_i = 0.01$, SA algorithm is applied to the new offspring.

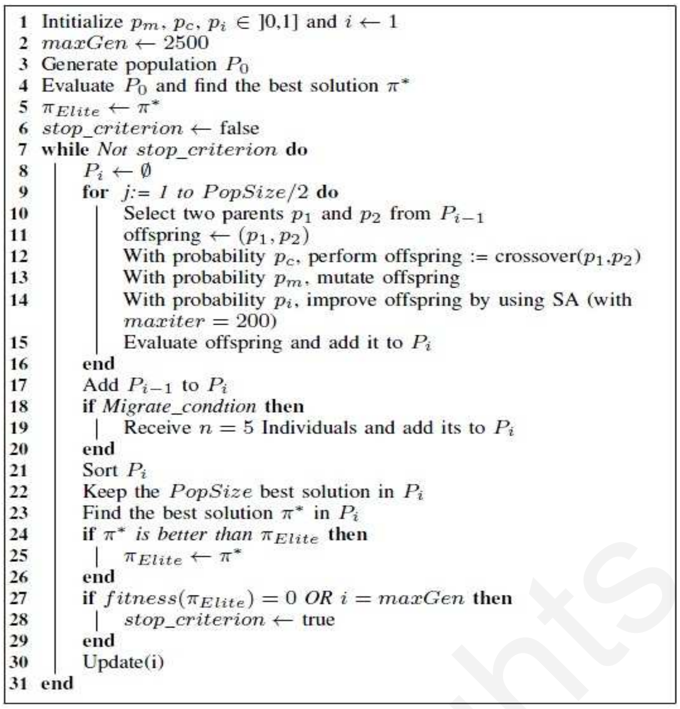
6.2 Our distributed Hybrid GA-SA Algorithm

The principles of our Distributed hybrid GA-SA (*GASA_Dist*) algorithm are described in Algorithm 2. We use independent subpopulations of individuals with their own fitness functions which evolve in isolation, except for an exchange of some individuals (migration). A set of $m = 30$ individuals is assigned to each of the P processors, for a total population size of $m * P$. The set assigned to each processor is its subpopulation. The processors are connected by an interconnection network with a ring topology. Initial subpopulations consist of a randomly constructed assignment created at each processor. Each processor, separately and in parallel, executes the *GASA_Seq* algorithm on its subpopulation for a certain number of generations. Afterwards, each subpopulation exchanges a specific number of individuals (migrants) with its neighbors. We exchange the individuals themselves, i.e. the migrants are removed from one subpopulation and added to another. Hence the size of the population remains the same after migration. The process continues with the separate evolution of each subpopulation for a certain number of generations. At the end of the process the best individual that exists constitutes the final assignment.

Copy Rights

7. EXPERIMENTAL RESULTS

For our experiments, we consider a memory architecture composed by an SPM, an instruction cache and a DRAM. Our energy model is based on the OTAWA framework (Cassé and Rochange, 2007) to collect information about number of accesses and on the energy consumption estimation tool CACTI (Wilton and Jouppi, 1996) in order to collect information about energy per access to each memory kind.



```
Intitialize $p_m, p_c, p_i \in ]0,1]$ and $i \leftarrow 1$
$maxGen \leftarrow 2500$
Generate population $P_0$
Evaluate $P_0$ and find the best solution $\pi^*$
$\pi_{Elite} \leftarrow \pi^*$
$stop\_criterion \leftarrow$ false
while $Not\ stop\_criterion$ do
    $P_i \leftarrow \emptyset$
    for $j$:= 1 to $PopSize/2$ do
        Select two parents $p_1$ and $p_2$ from $P_{i-1}$
        offspring $\leftarrow (p_1, p_2)$
        With probability $p_c$, perform offspring := crossover($p_1$,$p_2$)
        With probability $p_m$, mutate offspring
        With probability $p_i$, improve offspring by using SA (with
        $maxiter = 200$)
        Evaluate offspring and add it to $P_i$
    end
    Add $P_{i-1}$ to $P_i$
    if $Migrate\_condtion$ then
        Receive $n = 5$ Individuals and add its to $P_i$
    end
    Sort $P_i$
    Keep the $PopSize$ best solution in $P_i$
    Find the best solution $\pi^*$ in $P_i$
    if $\pi^*$ is better than $\pi_{Elite}$ then
        $\pi_{Elite} \leftarrow \pi^*$
    end
    if $fitness(\pi_{Elite}) = 0$ OR $i = maxGen$ then
        $stop\_criterion \leftarrow$ true
    end
    Update(i)
end
```

Algorithm 2. Distributed hybrid GA-SA algorithm running on each processor.

Our presented GA-SA algorithms and TS have been implemented with the C++ language on a PC Pentium (D), with a 3 GHz processor and 1 Gbyte of memory running under Windows XP Professional version 2002. For the GA-SA distributed version, we used a cluster of 4 PCs having the same characteristics and the MPICH2 (version 1.0.7) library for communication across the processes. The machines were running their normal daily loads in addition to our algorithms. Table 2 presents the benchmarks used. They also can be downloaded from (Benchmarks, 2010). Due to the lack of space, just the WB cache policy results are plotted. Same amounts are recorded for the WT mode ($E_{WTmode} \neq E_{WBmode}$).

Table 2. List of Benchmarks.

Benchmarks	Suite	Description
ShaCE	MiBench	The secure hash algorithm that produces a 160-bit message digest for a given input.
BitcountCE	MiBench	Tests the bit manipulation abilities of a processor by counting the number of bits in an array of integers.
FirCE	SNU-RT	Finite impulse response filter (signal processing algorithms) over a 700 items long sample.
JfdctintCE	SNU-RT	Discrete-cosine transformation on 8x8 pixel block.
AdpcmCE	Mälardalen	Adaptive pulse code modulation algorithm.
CntCE	Mälardalen	Counts non-negative numbers in a matrix.
CompressCE	Mälardalen	Data compression using lzw.
DjpegCE	Mediabenchs	JPEG decoding.
GzipCE	Spec 2000	Compression.
NsichneuCE	Wcet Benchs	Simulate an extended Petri net. Automatically generated code with more than 250 if-statements.
StatemateCE	Wcet Benchs	Automatically generated code.

We investigate the energy performances of our hybrid GA-SA algorithms when compared to TS. 30 different executions for each heuristic were performed and the best and average results were recorded. Both *GASA*_Seq and *GASA_Dist* always find the best solution and give similar results. We therefore call *GASA* the common value of their executions. Figure 1 presents the results obtained. Both GA-SA heuristics achieve

Copyrights

better performances than TS. In fact, they consume from 76.23% (StatemateCE) up to 98.92% (ShaCE) less energy than TS.

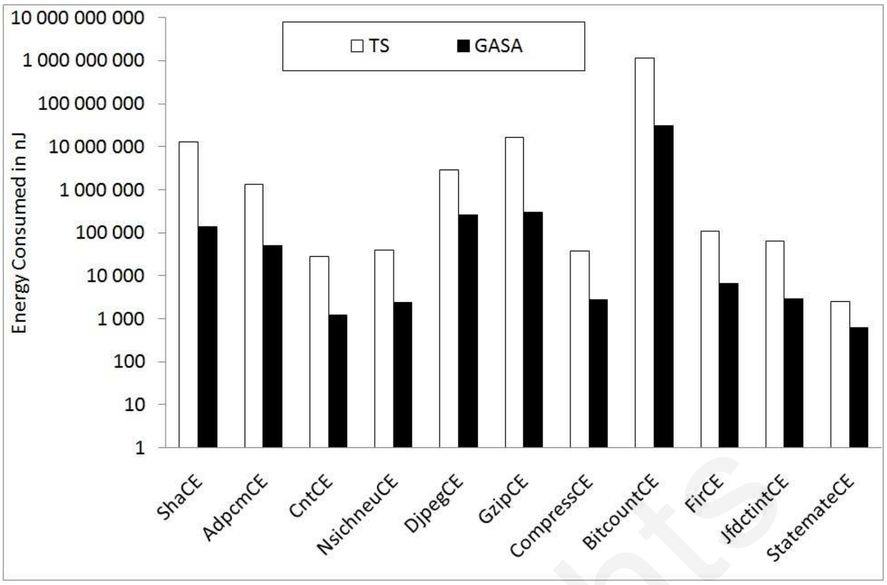


Figure 1. Energy consumed by our benchmarks with WB mode.

We also record the average execution times needed by *GASA_Seq* and *GASA_Dist* to achieve the 30 executions. Figure 2 presents the results obtained on the largest (size) benchmarks. From this figure, we can notice that the distributed GA-SA version (*GASA_Dist*) is faster than the sequential GA-SA version (*GASA_Seq*). In fact, *GASA_Dist* requires from 72.31% (GzipCE) up to 74.67% (CntCE) less execution time than *GASA_Seq*.

Copy Rights

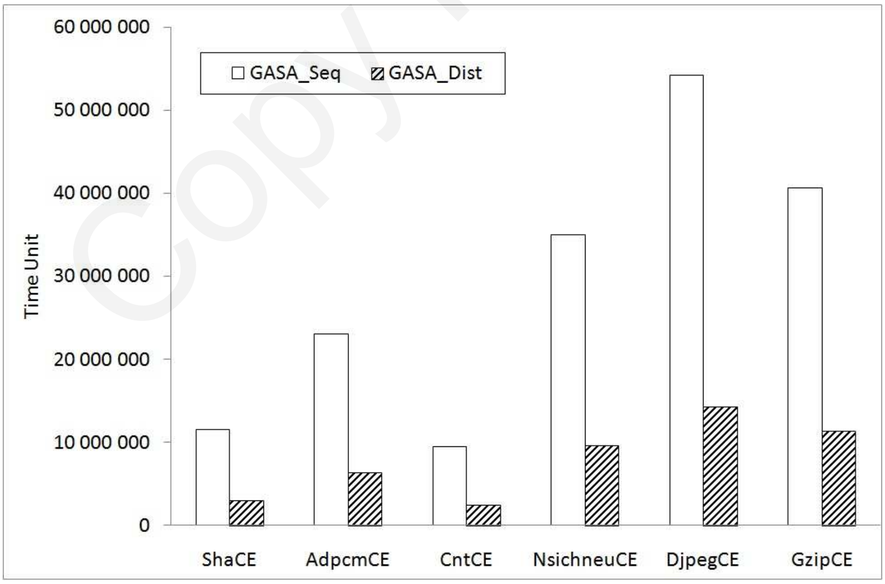


Figure 2. Execution time used by our GA-SA algorithms.

8. CONCLUSION AND PERSPECTIVES

We have proposed new hybrid sequential and distributed algorithms based on SA and GAs in order to reduce memory energy consumption in embedded systems. Our GA-SA algorithms consume nearly from 76% up to 98% less memory energy than TS. Execution time savings for the distributed GA-SA version (nearly from 72% up to 74% for a cluster of 4 PCs) are also recorded. In future work, we plan to investigate the same problem when relaxing the memory constraints by considering random memory sizes on one hand. On the other hand, we plan to explore evolutionary heuristics.

ACKNOWLEDGEMENT

The work of M. Idrissi Aouad and O. Zendra is supported by the french national research agency (ANR) in the Future Architectures program.

REFERENCES

Adiletta, M. et al, 2002. The Next Generation of Intel IXP Network Processors. *In Intel Technology Journal*, Vol. 6, No. 3, pp. 6-18.

Benchmarks, 2010. www.loria.fr/~idrissma/benchs.zip.

Ben Fradj, H. et al, 2005. Energy Aware Memory Architecture Configuration. *In ACM SIGARCH Computer Architecture News*, Vol. 33, No. 3, pp. 3-9.

Cassé, H. and Rochange, C., 2007. OTAWA, Open Tool for Adaptative WCET Analysis. *Proceedings of Design, Automation and Test in Europe (DATE)*. Nice, France. Poster session.

Idrissi Aouad, M. et al, 2010. A Tabu Search Heuristic for Scratch-Pad Memory Management. *Proceedings of International Conference on Software Engineering and Technology (ICSET)*. Rome, Italy, Vol. 64, pp. 386-390. WASET Publisher.

Idrissi Aouad, M. and Zendra, O., 2007. A Survey of Scratch-Pad Memory Management Techniques for low-power and -energy. *Proceedings of 2nd ECOOP Workshop on Implementation, Compilation, Optimization of Object-Oriented Languages, Programs and Systems (ICOOOLPS)*. Berlin, Germany, pp. 31-38.

ITRS, 2007. System Drivers. http://www.itrs.net/Links/2007ITRS/2007_Chapters/2007_SystemDrivers.pdf.

Kellerer, H. et al, 2004. *Knapsack Problems*. Springer Publisher, Berlin, Germany.

Kirkpatrick et al, 1983. Optimization by Simulated Annealing. *In Science*, Vol. 220, pp. 671-680.

Panda, P. R. et al, 1997. Efficient Utilization of Scratch-Pad Memory in Embedded Processor Applications. *Proceedings of European Design and Test Conference (EDTC)*. Washington, DC, USA, pp. 7.

Sivanandam, S. N. and Deepa, S. N., 2007. *Introduction to Genetic Algorithms*. Springer Publishing Company, Incorporated.

Tanenbaum, A., 2005. *Architecture de l'ordinateur 5^{e} édition*. Pearson Education.

Wehmeyer, L. et al, 2004. Compiler-Optimized Usage of Partitioned Memories. *Proceedings of the 3rd Workshop on Memory Performance Issues (WMPI)*. Munich, Germany, pp. 114-120.

Wilton, S. and Jouppi, N., 1996. Cacti: An Enhanced Cache Access and Cycle Time Model. *In IEEE Journal of Solid-State Circuits*, Vol. 31, pp. 677-688.

Copy Rights

A LUA VIRTUAL MACHINE FOR RESOURCE-CONSTRAINED EMBEDDED SYSTEMS

Alex de Magalhães Machado and Antônio Augusto Fröhlich
Laboratory for Software and Hardware Integration
Federal University of Santa Catarina
P.O.BOX 476, 88040900, Florianópolis, Brazil

ABSTRACT

Embedded systems and high-level languages usually belong to different worlds. It is not easy to fit a language runtime environment to a strongly constrained embedded platform. This porting mostly causes a loss of functionalities. High-level languages need large support from lower-level hardware or software so they can work. This paper describes our work to port the Lua Virtual Machine (LVM) to the Embedded Parallel Operating System (EPOS), making possible the execution of applications written in a high-level language such as Lua on embedded systems. The final system with support for all Lua libraries has less than 155 KB of size, which makes it suitable for a large amount of embedded systems. More than that, our tests showed that LVM runs faster on EPOS than on a Linux distribution.

KEYWORDS

Embedded System, Virtual Machine, High-Level Language, Lua

Copy Rights

1. INTRODUCTION

High-level languages provide a vast set of abstractions to ease the development of applications. These abstractions are only possible with a considerable runtime support by the operating system and the hardware. Virtual machines are increasingly providing this type of support, although they require considerably more resources than usual, often impacting applications performance beyond the acceptable. Embedded systems can hardly afford such resources (Koshy et al, 2009). Sensor networks, for example, are a domain characterized by resources that are orders of magnitude smaller than what ordinary virtual machines require (Levis and Culler, 2002).

Therefore embedded systems and high-level languages usually belong to different worlds. It is not easy to fit the runtime environment of a programming language to a strongly constrained embedded platform. This porting mostly causes a loss of functionalities (Caracas et al, 2009). Additionally, languages also have tools specially designed for general-purpose computers, and some abstractions are useless or less important in embedded systems, and their support could be simplified. Our ultimate goal is to understand the environment's behavior, and then remove any unnecessary overhead. Without that overhead, the use of high-level languages in embedded systems would become reality. We want to change this scenario, by analyzing all the abstractions between application and hardware, and trying to reduce the gap between them.

This work specifically describes the adaptations in the Lua Virtual Machine (LVM) for the Embedded Parallel Operating System (EPOS). Existing embedded virtual machines are usually a subset of a desktop virtual machine. These embedded virtual machines normally do not provide features that are hard to implement on resource-constrained embedded systems. This approach eventually impacts execution time, since the virtual machine was not previously designed for this environment (Caracas et al, 2009). However, we decided to work on Lua because the LVM is a lightweight virtual machine written in C, and it is portable as far as it concerns the hardware and operating system support for libc (Ierusalimschy et al, 2007). Nonetheless, the LVM normally requires support to some functionalities that are not used by any Lua application running on EPOS, such as software localization, modules loading, and access to environment variables and external commands. These features were removed. To give a shape to the remaining features, we defined a Lua profile, in which we create a set of available resources for the embedded version of LVM.

This Lua profile aims to help Lua developers to understand the functionalities LVM provides for embedded Lua applications.

The rest of the paper is organized as follows. Section 2 describes related works on interoperability between high-level languages and embedded systems. Section 3 describes the support required by LVM, and what EPOS could provide. It also describes how we solved the problems of functionalities LVM needed but EPOS could not provide. Section 4 summarizes our Lua Profile, and Section 5 presents the evaluation of our embedded LVM in terms of size and performance. Section 6 concludes the paper and describes future works.

2. RELATED WORK

High-level languages are becoming more popular in embedded systems because they provide useful programming abstractions such as object-orientation and multi-threading (Ishikawa and Nakajima, 2005). However, embedded systems are characterized by great architectural diversity. This can be tackled by virtual machines, at the expense of impacts in performance. Various techniques have been developed in order to reduce this performance gap between native and virtual execution environments (Koshy et al, 2009).

There are different approaches for virtual machines: virtualizing real hardware, virtualizing intermediate program representation and virtualizing bytecode interpretation (Costa et al, 2007). We are focusing our work in virtual bytecode interpretation. This set of virtual machines can be classified in two classes. The first class targets middleware by position between operating system and applications. They are called middleware level virtual machines. The second replaces the entire operating system. They are called system level virtual machines (Costa et al, 2007). Since LVM does not replace the operating system, it is a middleware level virtual machine. In fact, EPOS is our operating system.

High-level languages have abstractions that require special support from the virtual machine (Koshy et al, 2009), but using virtual machines in embedded systems has some pros and cons. They enable higher levels of abstraction, but they also have to care about the execution model. LVM has a straightforward execution model, which did not have to be changed. LVM is a register-based virtual machine, unlike most of the current middleware level virtual machines for bytecode interpretation. This type of virtual machine can be implemented to be faster than stack-based virtual machine. This approach has been rewarding, since programs require far less instructions in order to execute a task (Shi et al, 2005).

The Mote Runner virtual machine is an interesting approach because it was created from scratch and it supports more than one language (Caracas et al, 2009). However, its implementation removes some features from its supported languages, e.g. threads, floating point arithmetic, some data types, multi-dimensional arrays, and introduces the event-driven programming paradigm, hence causing a negative impact on the application portability. Mote Runner only supports strictly-typed languages, unlike our approach (Caracas et al, 2009). Our approach does also remove some features, but only those that would not be used on an embedded system, such as software localization, dynamic module loading, and access to environment variables.

There are other virtual machines, such as VM* and EarlGray, with different approaches on Java Virtual Machines. VM*, specifically, focuses on optimizing a Java Virtual Machine for wireless sensor networks (Koshy et al, 2009). However, most of its optimizations are already used in the LVM or change the language's functionalities, hence impacting its portability. EarlGray is a Component-based Java Virtual Machine, therefore focusing on modularization (Ishikawa and Nakajima, 2005). This modularization is achievable through the use of EPOS, considering LVM and its libraries as components (Schulter et al, 2007). Scylla and KESO are other virtual machines aiming efficiency and low overhead, but they have very different ways to solve these problems. In fact, Scylla is a system level virtual machine (Stanley-Marbell and Iftode, 2000), and KESO is a tool that compiles Java bytecode to C source code, the system's native language (Stilkerich et al, 2006).

Some of these approaches are unsuitable for small embedded devices, and others make significant changes to the supported languages, requiring applications to be designed specifically for the host system, therefore impacting portability.

3. EPOS SUPPORT FOR LVM

This section describes in details our approach to port the LVM to EPOS. LVM is written in C and compiles as a C++ library with minor changes (Ierusalimschy et al, 2007). That is how we use it on EPOS. Therefore, its portability relies only on the underlying hardware and operating system.

EPOS is a framework designed to guide and provide architecture transparency to the development of scenario independent component families that can be used in different environments through applying aspect programs (Schulter et al, 2007; Fröhlich and Schröder-Preikschat, 2000). It is designed for embedded hardware and it provides most of the support Lua needs to work. Nevertheless, Lua uses C standard libraries that are not present in EPOS. We could add these libraries to the environment and most of it would work properly, but we would be adding unnecessary code. We therefore chose to implement only the functionalities Lua needs. The next subsections address every functionality we had to handle in order to allow full support for Lua applications on EPOS.

3.1 Character, Memory, and String Handling

Character Classification functions are used in LVM inside lexical analyzer routines and also for pattern matching. Since these functions were not available in EPOS, they were implemented. This implementation was simple and took into consideration the ASCII character set.

EPOS already had a header called string.h, which defined some Memory and String handling functions. Lua uses most of these functions and a few more. Since they are frequently used, they were implemented inside the file string.cc. The only function that we did not implement was strcoll, although it was used in the LVM. The calls to this function were replaced by a call to the strcmp function. We will discuss more about this subject in the next subsection.

3.2 Localization, Time, and Date

The main use for software localization tools in the LVM is to provide software localization for the Lua applications. However, the current locale can also be used to inform the lexical analyzer what is the current representation for the decimal point. The Lua Operating System Library provides the setlocale function for Lua applications. This function simply calls the C standard library. We could not find a utility for software localization in EPOS, so we removed this feature. This removes the setlocale function from the Lua library. Besides, we removed the support for locale in every function that originally used it, such as String and Time handling functions.

The Lua Operating System Library provides the following manipulation and formatting functions to Lua applications: 'clock', 'date', 'difftime', and 'time'. Lua uses C library functions in order to achieve that. EPOS provides the following classes for these purposes: Real-Time Counter, Date, Clock, and Chronometer. We implemented these functions using the classes above. Our implementation differs from the original in that it does not take into account the current locale, as we already discussed.

Our Chronometer class counts how much time passed since the last call to its 'start' function. In order to do so, it uses an architecture-dependent function called 'time_stamp'. We used the Chronometer class to implement the 'clock' function. Our Real-Time Counter class uses its machine-dependent functions to implement some of the other functions.

3.3 General Purpose Standard Library

The C standard library is used in LVM for various purposes. The realloc function is used for all the LVM memory allocations. It was implemented in the file malloc.cc using the EPOS malloc function. The exit function is used when an error occurs. We now use the EPOS exit function of the Thread class. The strtoul, abs and strtod functions were not provided by EPOS and therefore they were implemented straightforwardly. The getenv and system functions were not implemented. The getenv function is mostly used for loading and building modules in Lua. Currently, the embedded lua virtual machine only executes one Lua script at a time, which is placed inside EPOS application. Therefore, the Lua Package Library was disabled. Besides this

Copy Rights

specific use, the Lua Operating System Library also provides these functions to Lua applications. They were not implemented because they use environment variables and external commands, and our Lua profile does not support these functionalities, as we will discuss further.

The Lua Mathematical Library provides the rand and srand functions for Lua applications. The first generates a random number and the second sets the seed. They were implemented in EPOS using the Pseudo_Random class with some minor changes.

3.4 Input and Output

Various file systems were developed for EPOS, but they are not compatible with the way general-purpose computers use file systems. The functions that receive or return files were not implemented. However, we created an FStream class, similar to the OStream class present on EPOS, and declared all those functions there. Hence these functions are not supported, but they could be eventually implemented and their support would be ready to work. The functions responsible for reading from standard input stream were not implemented, since EPOS has no input device by default. This also could be easily implemented the same way of the FStream class. The functions responsible for writing to standard output stream were implemented using the existing OStream class.

3.5 Mathematical Operations

Lua provides the Lua Mathematical Library for its applications. Nevertheless, the underlying hardware does not necessarily support floating-point number representation. We implemented a simple configuration which informs the virtual machine whether the hardware supports or not floating-point representation. If so, these functions would be available for Lua applications. If not so, the Lua Mathematical Library would be reduced to only fixed-point functions.

Copy Rights

4. LUA PROFILE

The set of modifications described in the last section only addresses C library functionalities in terms of what LVM needs and what EPOS provides. The work we did in adapting LVM to EPOS, implementing new features and changing others, accomplishes our goal of providing embedded system support for LVM.

Table 1. Lua Library functions. Crossed out words are not present on our Lua Profile, and thus are not available on EPOS

Basic library	Package Coroutine library	String library	Mathematical library	I/O library	OS and Table library	Debug library
assert , error	~~module~~	byte	abs	io.close	clock	debug
collectgarbage	~~require~~	char	acos , asin	io.flush	date	getfenv
~~dofile~~ , ~~loadfile~~	~~cpath~~	dump	atan , atan2	~~io.input~~	difftime	gethook
getfenv	~~loaded~~	find	ceil , floor	~~io.lines~~	~~execute~~	getinfo
getmetatable	~~loaders~~	format	cos , sin	~~io.open~~	exit	getlocal
ipairs , pairs	~~loadlib~~	gmatch	cosh , sinh	io.output	~~getenv~~	getmetatable
load , loadstring	~~path~~	gsub	deg	~~io.popen~~	~~remove~~	getregistry
next	~~preload~~	len	exp	~~io.read~~	~~rename~~	getupvalue
pcall , xpcall	~~seeall~~	lower , upper	fmod	~~io.tmpfile~~	~~setlocale~~	fenv
print		match	frexp , ldexp	~~io.type~~	time	sethook
rawequal	create	rep	log , log10	io.write	~~tmpname~~	setlocal
rawget, rawset	resume	reverse	max , min	~~file:close~~		setmetatable
select	running	sub	modf	~~file:flush~~	concat	setupvalue
setfenv	status		pow	~~file:lines~~	insert	traceback
setmetatable	wrap		rad	~~file:read~~	maxn	
tonumber	yield		random	~~file:seek~~	remove	
tostring			randomseed	~~file:setvbuf~~	sort	
type			sqrt	~~file:write~~		
unpack			tan , tanh			

Through these modifications, we defined features that EPOS would not support and consequently LVM cannot use. That is the case of OS library functions such as ‘getenv’, ‘execute’ and ‘setlocale’. The ‘execute’ function, specifically, is useful for Lua applications that control larger systems, but this kind of control can be performed outside Lua, with the use of EPOS facilities. We are looking forward to support this ‘execute’ function inside Lua in future works.

However, two problems arise from this scenario. The first is that Lua developers may not know whether EPOS supports an LVM functionality or not. The second is that Lua developers may never need some features in an embedded Lua application, but LVM would still support these features.

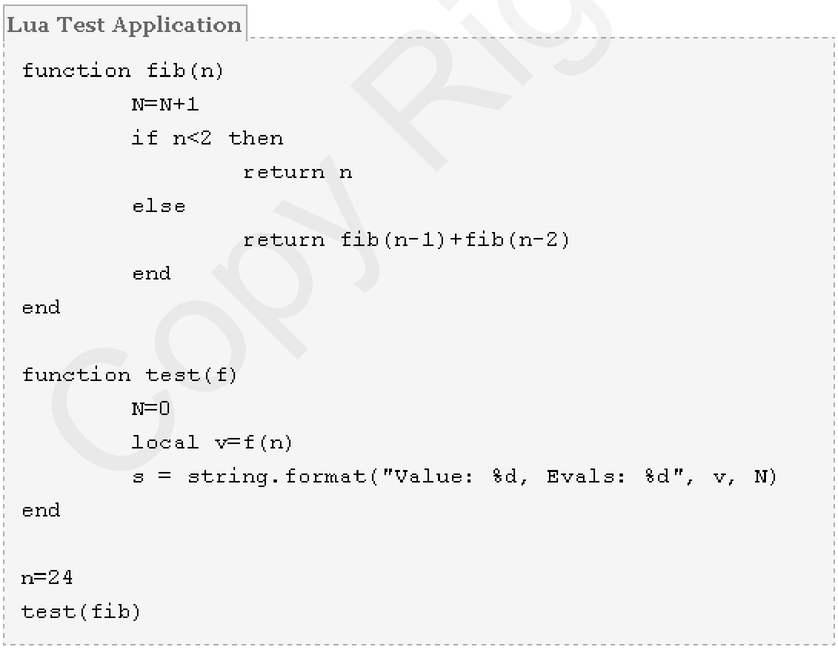
Therefore, in order to properly create an efficient and maintainable Lua runtime environment for embedded systems, we need to make clear what LVM provides and what Lua applications need. Even if a feature is supported by EPOS and LVM, it is possible that it will never be used by a Lua application running on EPOS. We solve these two problems creating a Lua Profile for embedded systems, which defines a subset of functionalities that LVM provides for embedded Lua applications.

Table 1 shows all Lua libraries, with all the functions they implement. The features removed in our Lua Profile are crossed out. As we can see, we did not remove many features from the LVM, and therefore our Lua Profile is vast and it is able to support most of the real Lua applications.

5. EVALUATION

We intented to evaluate the overhead LVM caused on EPOS environment in terms of size, and also analyse if the embedded LVM runs on EPOS similarly to a Linux distribution, so Lua developers could expect comparable performance between them.

Copy Rights



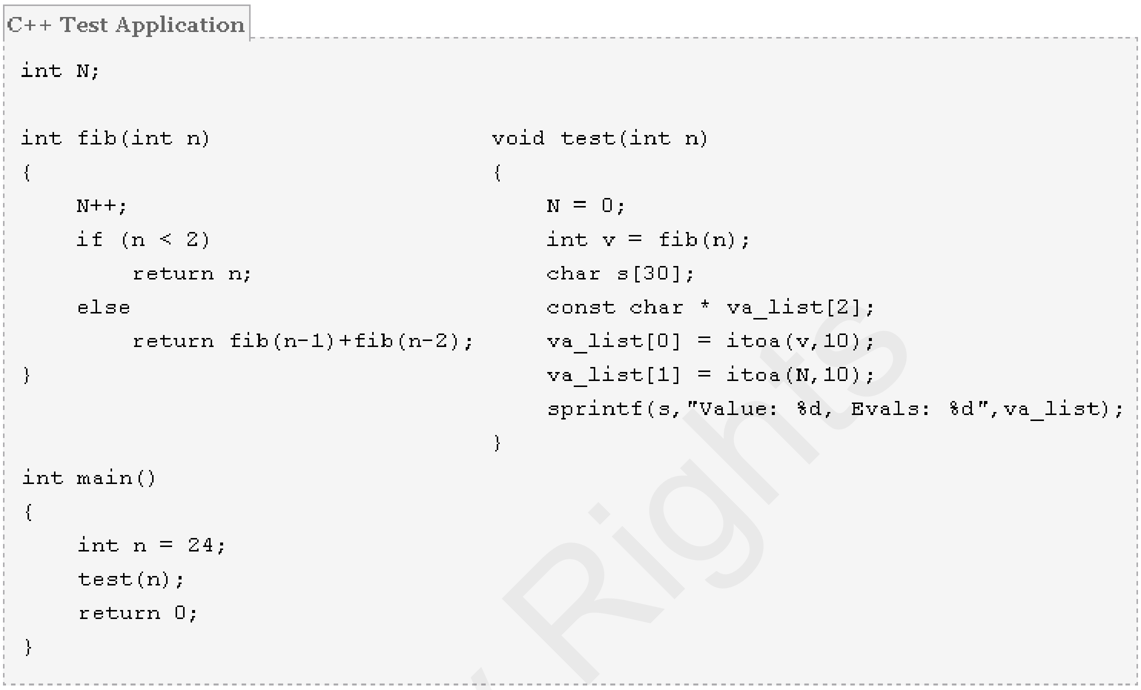
Lua Test Application

```
function fib(n)
        N=N+1
        if n<2 then
                return n
        else
                return fib(n-1)+fib(n-2)
        end
end

function test(f)
        N=0
        local v=f(n)
        s = string.format("Value: %d, Evals: %d", v, N)
end

n=24
test(fib)
```

Figure 1. Lua test application.

We created a test application in Lua and in C++, EPOS native language. Fig. 1 and Fig. 2 show the source code of the two versions. This simple application calculates the 24th Fibonacci number. An enhanced Lua version of this test application is deployed with the source code of the official implementation of Lua. The Lua version uses a recursive function called fib and then calls the format function of the String library to create a result string. The C++ version uses our own implementation of functions itoa and sprint. We tested these applications on EPOS and on an Ubuntu 9.04 with 2.6.28-19 kernel. Both systems were executed on the same platform. Therefore, we have four different execution times. All these execution times were measured with an oscilloscope.

As we can see in Fig. 3, the C++ application is faster than the Lua application wherever it is running. Nonetheless, Lua is still very attractive, since its high-level abstractions ease the development of applications. The C++ application was faster on the Linux distribution because of optimizations in the libc. The Lua application, however, was faster on EPOS.



C++ Test Application

```
int N;

int fib(int n)
{
    N++;
    if (n < 2)
        return n;
    else
        return fib(n-1)+fib(n-2);
}

void test(int n)
{
    N = 0;
    int v = fib(n);
    char s[30];
    const char * va_list[2];
    va_list[0] = itoa(v,10);
    va_list[1] = itoa(N,10);
    sprintf(s,"Value: %d, Evals: %d",va_list);
}

int main()
{
    int n = 24;
    test(n);
    return 0;
}
```

Copy Rights

Figure 2. C++ test application.

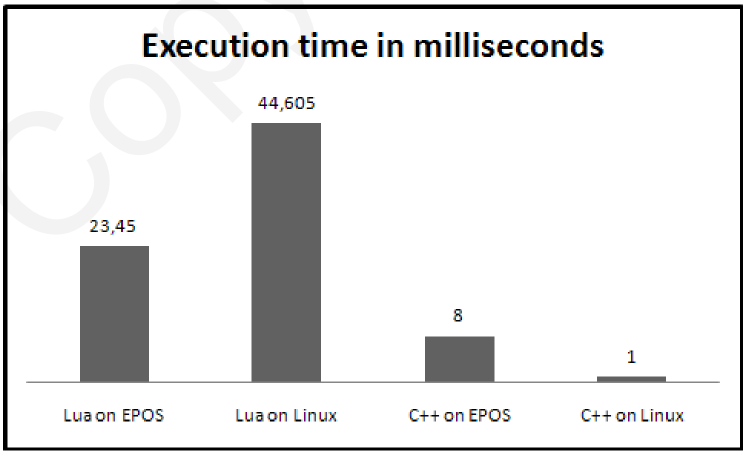


Figure 3. Execution time in milliseconds of the two applications running on the two different systems.

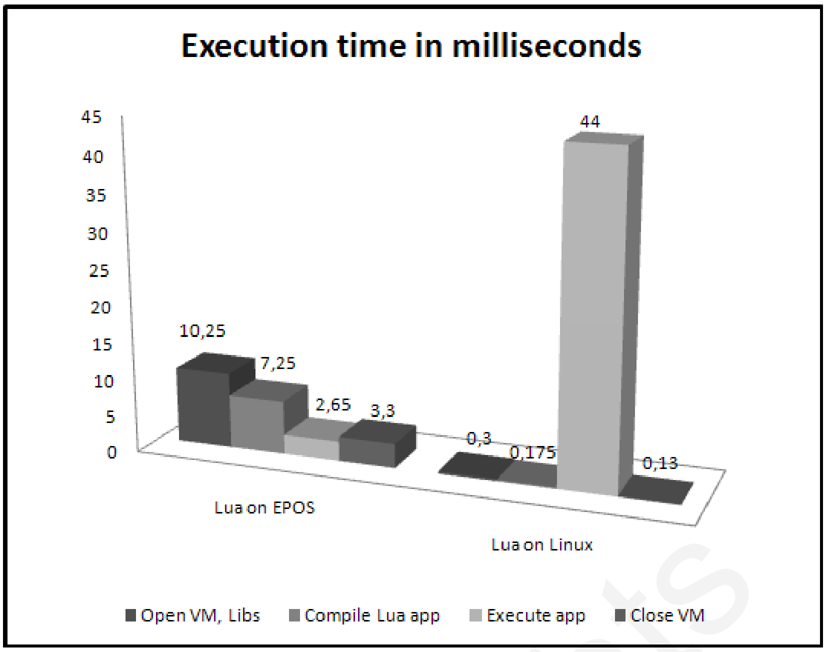


Figure 4. Execution time in milliseconds of each step of the LVM execution on EPOS and Linux.

Fig. 4 goes beyond and shows how much time it takes to perform each step of the LVM execution. These discrepant times depend primarily on the application, and therefore we cannot say that Lua runs faster or slower on EPOS. In fact, we are showing that the overhead of the LVM execution on EPOS is not much different from the Linux version. For the record, the LVM execution time standard deviation was 0.8857 ms, and the application execution time specifically had 0.1985 ms of standard deviation.

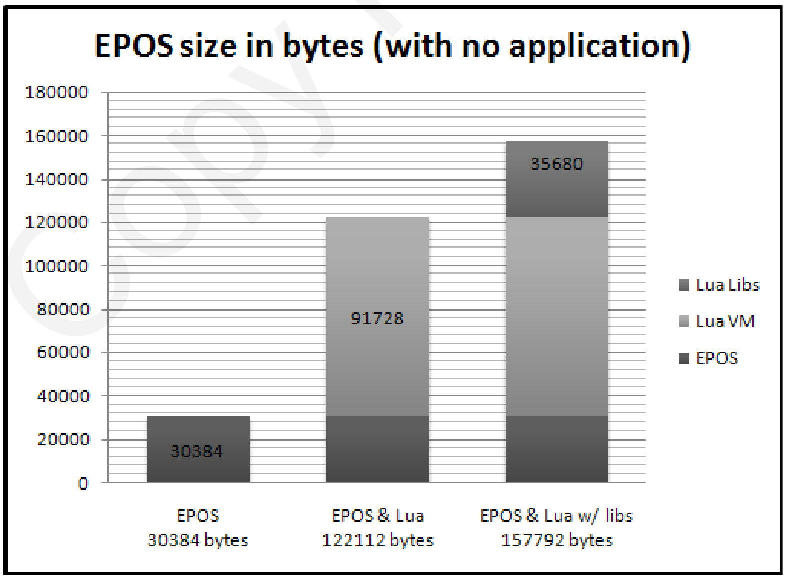


Figure 5. EPOS size in bytes, without any application.

Fig. 5 shows that EPOS has less than 32KB of size. The LVM, without its libraries, has size of approximately 90KB. Hence the EPOS size with basic Lua support is approximately 120KB. The libraries size is approximately 35KB. Theses sizes do not count the application size, which varies depending on the system.

Copy Prints

6. CONCLUSION

Although some embedded systems often do not have 120KB available just to support the system, this size is comparable to those of our related works. In fact, some other high-level languages virtual machines have much bigger sizes (Ishikawa and Nakajima, 2005). Also, the LVM code in EPOS is really lighter than in Linux, but our tests show that we can improve performance and also reduce even more its size. That shall be the next step.

Our final EPOS/LVM system is able to execute a high-level language application, with full support to all high-level abstractions Lua already provided for desktop applications. More than that, we achieved to support a high-level language without restricting its flexibility, without disabling its portability, and most of all, without making the system unsuitable for embedded systems.

Finally, our approach showed us common aspects in adapting high-level languages for embedded systems, such as internal communications between virtual machine and operating system and libraries support by the virtual machine. We are looking forward to generalize these steps in order to ease the adaptations of virtual machine high-level languages for embedded systems.

REFERENCES

Caracas, A. et al, 2009. Mote Runner: A Multi-language Virtual Machine for Small Embedded Devices. *Sensor Technologies and Applications. SENSORCOMM '09. Third International Conference on*, pp. 117-125.

Costa, N. et al, 2007. Virtual Machines Applied to WSN's: The state-of-the-art and classification. *Systems and Networks Communications. ICSNC 2007. Second International Conference on*, pp.50-50.

Fröhlich, A. A. and Schröder-Preikschat, W., 2000. Scenario Adapters: Efficiently Adapting Components. *In Proceedings of the 4th World Multiconference on Systemics, Cybernetics and Informatics*. Orlando, USA.

Ierusalimschy, R. et al, 2007. The evolution of Lua. *In HOPL III: Proceedings of the third ACMSIGPLAN conference on History of programming languages*. New York, NY, USA, pages 2–1–2–26. ACM Press.

Ishikawa, H. and Nakajima, T., 2005. EarlGray: A Component-Based Java Virtual Machine for Embedded Systems. *Object-Oriented Real-Time Distributed Computing. ISORC 2005. Eighth IEEE International Symposium on*, pp. 403-409.

Koshy, J. et al, 2009. Optimizing Embedded Virtual Machines. *Computational Science and Engineering. CSE '09. International Conference on*, pp. 342-351.

Levis, P. and Culler, D., 2002. Maté: a tiny virtual machine for sensor networks. *In International Conference on Architectural Support for Programming Languages and Operating Systems*, pp. 85–95.

Schulter, A. et al, 2007. A Tool for Supporting and Automating the Development of Component-based Embedded Systems. *in Journal of Object Technology*, Vol. 6, No. 9, Special Issue: TOOLS EUROPE 2007, p. 399-416.

Shi, Y. et al, 2008. Virtual machine showdown: Stack versus registers. *ACM Transactions on Architecture and Code Optimization (TACO)*, Vol. 4, No. 4, p. 1-36.

Stanley-Marbell, P. and Iftode, L, 2000. Scylla: a smart virtual machine for mobile embedded systems. *Mobile Computing Systems and Applications, Third IEEE Workshop on*, pp. 41-50.

Stilkerich, M. et al, 2006. OSEK/VDX API for Java. *In Proceedings of the 3rd workshop on Programming languages and operating systems: linguistic support for modern operating systems*. San Jose, California, p.4-es.

Copy Rights

SUPPORTING INTERMOLECULAR INTERACTION ANALYSES OF FLEXIBLE-RECEPTOR DOCKING SIMULATIONS

Ana T. Winck, Karina S. Machado, Osmar Norberto de Souza and Duncan D. Ruiz
GPIN - Grupo de Pesquisa em Inteligência de Negócio
LABIO - Laboratório de Bioinformática, Modelagem e Simulação de Biossistemas
PPGCC, Faculdade de Informática, PUCRS,
Av. Ipiranga, 6681 – Prédio 32, sala 640, 90619-900, Porto Alegre, RS, Brazil

ABSTRACT

Nowadays, with the growth of biological experiments, solving and analyzing the massive amount of data being generated has been one of the challenges in bioinformatics. One important research area in bioinformatics is the rational drug design (RDD), centered on examining possible interactions between receptors and ligands, usually performed by molecular docking. Very little is known about the effectiveness of considering the flexibility of a receptor molecule in current molecular docking simulations when this flexibility is modeled by a molecular dynamics (MD) simulation. This methodology generates vast amounts of data which need to be explored in order to generate useful information about receptor-ligand interactions. To better interpret these data we developed a comprehensive repository that integrates features of all snapshots from a MD simulation trajectory with related data about receptor-ligand interactions from the docking simulations results. The prepared and stored data allowed the identification of residues that interact often with the tested ligands and that would not be seen in docking simulations with a rigid structure. We ranked, for each ligand, the top 10 receptor residues that interact in most of the docking runs, totalizing in 25 distinct residues for our receptor. These information would be arduous to obtain without our repository, as well as it cannot be assessed using a rigid receptor conformation, showing the importance of the receptor flexibility in molecular docking simulations.

KEYWORDS

Data Repository, Drug Design, Flexible Receptor, Molecular Docking

1. INTRODUCTION

Advances in molecular biology and the intensive use of computer modeling and simulation tools over the past years have had a deep impact in the drug discovery process (Lengauer and Rarey, 1996), turning viable the rational drug design (RDD) (Kuntz, 1992). In RDD there are basically receptors and ligands, where receptors are proteins that recognize and bind a compound. Ligands are small molecules with biological activities able to inhibit the target protein activity (Lesk, 2002). The *in-silico* based RDD is a four-step cycle centered on examining possible interactions between receptors and ligands, usually performed by molecular docking software (Lybrand, 1995). During the molecular docking a large number of simulations are performed in order to identify the ligand's best fit in the receptor. Most docking simulation software can deal with the ligand flexibility. However limitations occur when it is necessary to also consider the receptor flexibility (Totrov and Abagyan, 2008) (Cozzini et al, 2008). These limitations are specially challenging because of the large number of degrees of freedom that a receptor macromolecule owns.

It is highly desirable to take into account the receptor flexibility during molecular docking simulations (Cozzini et al, 2008) because, normally, the receptor can modify its shape upon ligand binding, molding itself to be complementary to its ligand. These conformational changes increase favorable contacts and reduce adverse interactions, resulting better total free energy of binding - FEB (Verkhiver et al., 2002). Moreover, state of the art docking algorithms predict an incorrect binding pose for about 50-70% of all ligands when only a single, rigid receptor conformation is considered (Totrov and Abagyan, 2008).

There are several approaches to address the receptor flexibility in molecular docking simulations, as revised by (Totrov and Abagyan, 2008) (Cozzini et al., 2008) (Alonso et al., 2006). In this work we address the explicit flexibility of the receptor by performing a series of molecular docking simulations considering, in each one of them, a different conformation or snapshot of the receptor, generated by molecular dynamic (MD) simulation (van Gunsteren and Berendsen, 1990). An important drawback when executing these docking simulations is the large amount of data generated, that makes impractical expert analyses on them. Depending on the MD simulation time scale, the data volume can increase exponentially.

A careful analysis of flexible-receptor molecular docking results, particularly those related to details of receptor-ligand interactions, is essential to improve the process of docking as a whole. Such analysis can be better explored if we integrate data from both MD simulations and flexible-receptor docking simulations. This integration can, furthermore, provide an easy manner to retrieve and preprocess data to be analyzed.

Considering that the increase in data volume can be unpredictable, key issues and challenges, such as persistence, fast retrieval and easy access to data and their provenance must be addressed. For these reasons, we felt encouraged to propose a comprehensive data repository to store all characteristics involved in MD simulations snapshots and flexible-receptor docking in a detailed level, e.g. the atom's coordinates involved in each molecular docking result. Such repository can be viewed as a powerful infrastructure for data staging. A preliminary conceptual model of this repository was previously introduced in (Winck et al., 2009).

In this article we present a detailed description of our repository model and its stored data. Having this repository at hands, with all our MD and docking results integrated, it made possible to perform biophysical analyses of flexible receptor-ligand interactions that would be arduous or impossible to perform without such support. Such analyses help us to better understand the role of receptor flexibility in docking simulations. By retrieving the stored data we were able to identify the receptor residues that interact in a receptor-ligand complex, where such kind of information cannot be assessed using a single, rigid receptor conformation.

2. THE PROPOSED REPOSITORY

We developed our repository aiming at integrating MD simulations and flexible-receptor docking results. In doing so, the proposed repository model is comprehensive enough to store all the provenance information about MD simulation and their generated snapshots integrated to molecular docking features. The repository model contains 17 tables. These tables, as well as their most important fields, are shown in Figure 1. Table 1 summarizes the main features of whole tables.

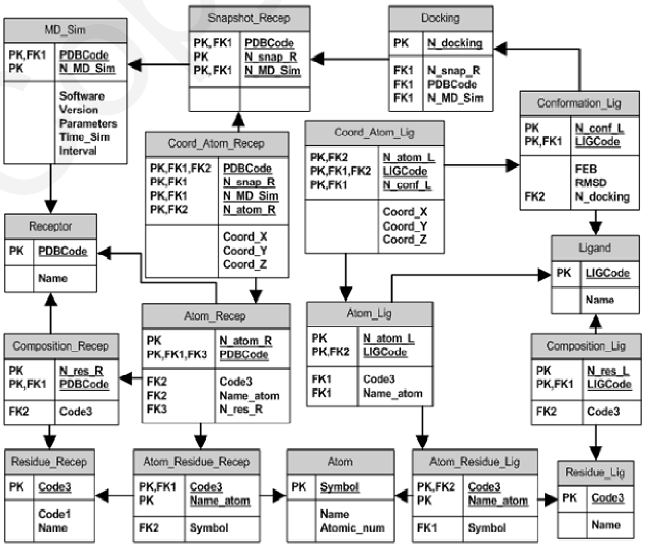


Figure 1. Conceptual model of our repository built using Microsoft Visio design tool.

Our repository allows us to retrieve spatial and temporal characteristics about the receptor and ligands stored since their conformations are described in terms of their time execution and of the spatial coordinates

Copy Rights

of each atom. The main advantage of this model is that each receptor and ligand can be analyzed in terms of their atoms, where each atom is related with the residues they are part of, even if they are stored in distinct tables. In doing so, we can make queries using both top-down and bottom-up approaches, addressing adequately ligands conformations and receptor snapshots's information on different views.

Table 1. Description of the 17 tables.

Table Name	Description
Atom	All periodic table atoms
Atom_Lig	All atoms of a given ligand
Atom_Recep	All atoms of a given receptor
Atom_Residue_Lig	All relationships between ligand's residues and ligand's atoms
Atom_Residue_Recep	All relationships between receptor's residues and receptor's atoms
Composition_Lig	All the relationships between the ligand's residues and the ligands
Composition_Recep	Relationships between receptor's residues and the 20 natural amino acids
Conformation_Lig	Each run of each performed molecular docking
Coord_Atom_Lig	Spatial coordinates of ligand atoms
Coord_Atom_Recep	Spatial coordinates of receptor atoms
Docking	Docking execution features
Ligand	Ligand details like number of atoms and name
MD_Sim	MD simulation provenance data
Receptor	Receptor details, like PDB header
Residue_Lig	All the ligands' residues
Residue_Recep	Three-letter code of the 20 natural amino acids
Snapshot_Recep	Snapshot details, like its time in the MD simulation trajectory

2.1 The Data stored in the Repository

To test our proposed model, in this work we consider one receptor and four ligands. However, it is important to keep in mind that, as can be seen in Figure 1, our model is able to store as many receptors and ligands as necessary. The receptor considered here is the InhA enzyme from *Mycobacterium tuberculosis* (Mtb) (Dessen et al., 1995). Four ligands were considered: nicotinamide adenine dinucleotide (NADH) (Dessen et al., 1995), triclosan (TCL) (Kuo et al., 2003), pentacyano(isoniazid)ferrate II (PIF) (Oliveira et al., 2004) and ethionamide (ETH) (Banerjee et al., 1994).

The flexibility of the InhA enzyme was obtained from three different MD simulations. The first one has 3,100 ps (1.0 ps = 10^{-12}s) considering Normal Pressure and Temperature (NPT) at 298 K and was performed using the software AMBER6 (Pearlman et al., 1995) as previously described in (Schroeder et al., 2005). The second and third MD simulation trajectories have 20,000 ps each considering NPT at 298 K and at 310 K, respectively and were performed using the AMBER9 (Case et al., 2006) as presented in (Gargano, 2009). The ligand files were obtained from the ZINC (Irwin and Soichet, 2005) database of small molecules in a MOL2 format.

We performed flexible-receptor docking simulations considering the first MD simulation trajectory (3,100 ps) and four ligands (Machado et. al, 2007). Each of the four ligands were submitted to a 3,100 docking simulations, where in each experiment one different receptor snapshot was used. The current population of some tables is summarized in the Tables 2a (receptor data) and 2b (ligands data). In these tables we show the total number of atoms (column 2), the total number of receptor snapshots and ligands conformations (column 3) and the total number of atomic coordinates (column 4).

Table 2. Current amount of receptor data stored (a) and current amount of ligand data stored (b).

MD	Table names in the repository		
Simulations	Atom_Recep	Snapshot_Recep	Coord_Atom_Recep
1	4,008	3,100	12,424,800
2	4,008	20,000	80,160,000
3	4,008	20,000	80,160,000
Total		43,100	172,744,800

(a)

Ligand	Table names in the repository		
	Atom_Lig	Conformation_LIg	Coord_Atom_Lig
NADH	52	31,000	1,612,000
PIF	24	30,420	730,080
TCL	18	28,370	510,660
ETH	13	30,430	395,590
Total		120,220	3,248,330

(b)

Copy Rights

With regard to the InhA receptor, which contains 268 residues and 4,008 atoms, we have a total of 12,424,800 atomic coordinates' records for the first MD simulation trajectory. For the second and third MD simulation trajectories we have a total of 80,160,000 records for each one. This totalizes more than 172 million of atomic coordinates' records for the InhA receptor. Regarding the ligand's data, in Table 2b we have, for NADH, 31,000 different conformations. As this ligand has 52 atoms, it totalizes 1,612,000 atomic coordinates. The other ligands do not have 31,000 conformations because the docking simulations did not always converge for valid results. Altogether the four ligands generated 3,248,330 records. In spite of handling data from one enzyme and four ligands, we believe this data is large enough to test our repository.

3. ANALYSIS OF THE REPOSITORY DATA

By performing analysis on our repository data, we intend to show how we can obtain patterns from receptor-ligands interactions. The analysis performed in this work aims at identifying which are the receptor residues that interact with the ligands in the majority of the docking runs. We intend to use the results of this analysis to select receptor snapshots as well as to select new ligands to be tested.

Algorithm 1: Calculating interaction between receptor and ligands

```
 1.
 2.
 3.
 4.
 5.
 6.
 7.     End for
 8.    End for
 9.
10.
11.    End if
12.
13.
14.    End if
15.   End for
16.  End for
```

Copy Rights

In order to retrieve proper features to be analyzed we need to combine the entire coordinates' records of the receptor with the entire coordinates' entries of each ligand obtaining the minimum distances between atoms in the receptor´s residues and ligands, for each docking result of each four InhA-ligand complexes. That is, for each receptor (R) residue we calculate the distance between their atoms and the atoms of the ligand (L). We define 1 when there are interactions (minimum distance less than 4.0 Å) and 0 when there are no interactions (minimum distance greater than 4.0 Å) as described by Algorithm 1, which is composed by:

- Total_conformations$_L$ corresponds to the total number of conformations of a given Ligand considering all its runs docking execution with all the receptor snapshots.
- Total_Residues$_R$ stores the total number of residues on Receptor
- Residue$_R$ is a given receptor residue r
- Total_atoms_Res$_R$ stores the total number of atoms of Residue$_R$
- x_R, y_R and z_R correspond to spatial coordinates of each atom in Residue$_R$
- Ligand$_L$ is a given ligand
- Total_atoms_Lig$_L$ is the total number of atoms of Ligand$_L$
- t, r, i, j are variables used to control *For* loops
- x_L, y_L and z_L are the spatial coordinates of each atom in Ligand$_L$
- [DistM$_{i,j}$] stores all the distances between a given receptor residue and a given ligand
- [RESULT$_{t,r}$] stores all the results in a binary format

To illustrate, we present a distance matrix [$DistM_{i,j}$] for the GLY95 residue of the InhA receptor and the PIF ligand (suppose that $Total\_conformations_L$ and $Total\_Residues_R$ are fixed as 1). As GLY95 residue has 7 atoms, we have $Total\_atoms\_Res_R$ = 7 which means the 7 lines on the [$DistM_{i,j}$] matrix. PIF ligand has 24 atoms, so, $Total\_atoms\_Lig_L$=24 which correspond to the 24 columns on the [$DistM_{i,j}$] matrix (we do not show all the columns). It is important to mention that this matrix with 168 elements (7x24) is the result of the distance matrix for only one receptor residue with one ligand. The [$RESULT_{t,r}$] described above is only a part of the final [$RESULT_{t,r}$] matrix for the PIF ligand. In this example, columns 1, 95, 164, 195 e 268 correspond to the receptor residues 1, 95, 164, 195 and 268. The lines we describe on [$RESULT_{t,r}$] corresponds to the ligand conformations 1, 2, 391 and 30,420, respectively.

So, from each [$DistM_{i,j}$] we obtain just one value for the final matrix [$RESULT_{t,r}$]. To generate a proper input file with the results of all the receptor residues and a given ligand, performing the Algorithm 1 we have a combination of 12,424,800 coordinate's records of the receptor with the 730,080 coordinate's entries for the ligand PIF. It means that we need to generate 268 tables like the [$DistM_{i,j}$]. The results are arranged in 270 attributes: the first two are the serial number of the receptor snapshot and ligand conformation; the next 268 correspond to the values of [$RESULT_{t,r}$]. Their records, in a binary format, indicate if there is or not a residue receptor-ligand interaction. Table 3 illustrates this arrangement for the flexible InhA-PIF complex. This file has 30,420 instances, corresponding to the total of PIF conformations.

$$DistM_{i,j} = \begin{bmatrix} 7.78 & 7.77 & 5.99 & 5.76 & \dots & 4.22 & 5.83 & 5.73 & 7.77 \\ 8.44 & 8.45 & 6.21 & 5.80 & \dots & 4.65 & 6.50 & 6.44 & 8.06 \\ 6.50 & 6.87 & 5.80 & 5.58 & \dots & 5.50 & 5.62 & 7.02 & 7.70 \\ 7.12 & 7.16 & 6.66 & 6.55 & \dots & 3.81 & 5.95 & 7.72 & 8.44 \\ 5.82 & 5.52 & 4.91 & 4.84 & \dots & \mathbf{2.72} & 4.46 & 4.66 & 6.09 \\ 7.35 & 7.20 & 6.18 & 5.66 & \dots & 3.19 & 6.45 & 6.77 & 7.01 \\ 8.04 & 6.20 & 5.47 & 5.59 & \dots & 6.99 & 7.31 & 7.22 & 7.57 \end{bmatrix} \quad RESULT_{t,r} = \begin{bmatrix} 0 & \dots & 1 & \dots & 1 & \dots & 0 & \dots & 0 \\ 0 & \dots & 1 & \dots & 1 & \dots & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 1 & \dots & 0 & \dots & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 1 & \dots & 1 & \dots & 1 & \dots & 0 \end{bmatrix}$$

Table 3. Part of the PIF file used for analysis

Snapshot Conformation	Ligand Conformation	...	LYS 164	...	THR 195	...	LEU 268
1	1	…	1	…	0	…	0
1	2	…	1	…	1	…	0
…	…	…	...	…	…	…	…
94	391	…	0	…	1	…	0
…	…	…	…	…	…	…	…
3,100	30,420	…	1	…	1	…	0

Copy Rights

4. RESULTS AND DISCUSSION

The objective of this work is to investigate the importance of a receptor explicit flexibility in its intermolecular interactions with small molecules or ligands. The intermolecular interactions were evaluated through docking simulation for the particular InhA enzyme receptor from *M. tuberculosis* and four ligand molecules: NADH, PIF, TCL, and ETH. The inventory of these interactions was deposited in the repository we developed for this purpose (Section 2). This repository allows total integration between the receptor snapshots or conformations from a MD simulation trajectory, and its docking simulation results to the four ligands. With the support of the repository we now concentrate in identifying the InhA residues, considering its flexible-receptor model, that interacts the most with the four ligands investigated. Are the residues the same as the one encountered for the crystal structure? Do other InhA flexible-receptor model (F_InhA) residues also interact with the tested ligands? Table 4 summarizes the results that answer these questions. The Swiss-PDBViewer (Guex and Peitsh, 1997) was used for the rigid InhA (R_Inha)-ligand interaction analyses.

It is clear-cut from Table 4 the importance the explicit receptor flexibility can play in intermolecular interaction and recognition. While in the inflexible crystal structure R_InhA the NADH ligand interacts with 22 amino acids residues, in the flexible receptor model F_InhA it interacts with 185 amino acids residues. The differences in the number of receptor's residues interacting with ligands, in both F_InhA and R_InhA, are also striking for PIF, TCL, and ETH. Curiously, the smaller the ligand (Table 2b) the less it interacted with the F_InhA, although the same trend is observed for R_InhA. This is somewhat expected as smaller ligands are usually less flexible and have a smaller accessible surface area.

Table 4. Intermolecular interaction analyses in the flexible-receptor model of InhA (F_InhA) and the rigid, crystal structure (R_InhA, PDB ID: 1ENY)

Ligand	F_InhA-Ligand Interactions	R_InhA-Ligand Interactions
NADH	185	22
PIF	165	13
TCL	139	12
ETH	105	8

By summing all cells filled with 1 in [RESULT_(t,r)] we calculated how many times (the occurrence) each of the 268 amino acids residues of the F_InhA receptor interacted with each of the four ligands. For each ligand we ranked the F_InhA receptor residues that interact in most of the docking runs and selected the top 10. These top 10 residues are displayed in Figure 2. In Table 5, the top 10 residues for each ligand are highlighted. Their intersection (∩) of the top 10 residues for the ligands gives a total of 25 distinct residues.

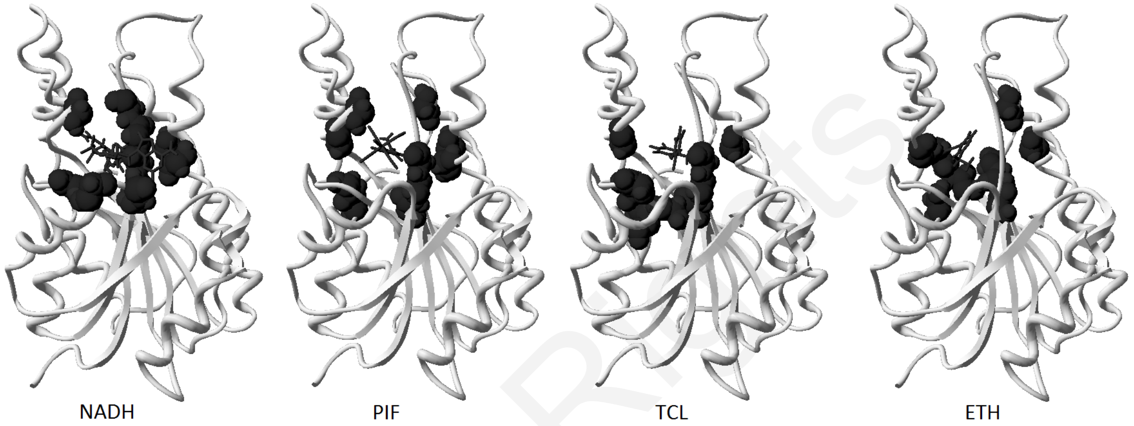


Figure 2. Top 10 amino acids residues of the F_InhA receptor that interact the most with the NADH, PIF, TCL, and ETH ligands. The InhA receptor (PDB ID: 1ENY) is represented by a ribbon model of its main chain. The top 10 residues are represented by van der Waals spheres, and the ligands by stick models.

For NADH, only five out of the top 10 residues are amongst the 22 found in the R_InhA-NADH interactions. The remaining five residues do not make contact with NADH in the crystal structure (PDB ID: 1ENY). Furthermore, after the intersection of the top 10 residues for all four ligands (Table 5), out of 25 distinct amino acids residues only 15 match their equivalents in the R_InhA receptor. Ten completely different residues are making contact with F_InhA. Hence, even for NADH, the coenzyme without which InhA does not function, flexibility seems to play an important role in mediating the equilibrium state of the InhA-NADH complex. In fact, a single ILE21 mutation in InhA turns *M. tuberculosis* resistant to isoniazid (INH), the major drug used to treat tuberculosis (TB) (Basso et al., 1998). It is the INH-NADH adduct that inhibits InhA (Rozwarski et al., 1998). Schroeder et al. (2005) showed that the inherent flexibility of InhA is affected by this mutation. As a consequence, NADH dissociates more readily from InhA, thus reducing its affinity for the INH-NADH inhibitor.

Seven of the top ten receptor residues that interact with PIF and TCL are the same (Table 5). This could be happening because these ligands are expected to bind to a similar region of the InhA receptor, the substrate binding site. However, for the 25 distinct residues we notice that many bind more often to PIF than to TCL, suggesting that, in fact, their binding site might be different regions in the InhA receptor. While TCL binds weakly to the substrate binding site of InhA (Kuo et al., 2003) and cannot be used as an effective drug against TB, PIF, which is based on isoniazid, binds very strongly to InhA. However, it is at an early stage of development (Oliveira et al., 2006) and its exact binding site on InhA is not known yet. We believe these results may help in the efforts to uncover the PIF binding site in the InhA receptor.

PIF and TCL are almost twice as big as ETH (Table 2b). Consequently, they cannot probably fit in the same region of InhA. Regarding ETH (Table 5), the selected residues make up part of the InhA pocket to which ETH should bind. It is important to point out that ETH is a pro-drug. It binds InhA as a covalent ETH-NADH adduct (Wang et al., 2007). As ETH is a small ligand, composed of 13 atoms (Table 2b), its binding region is different from the other ligands. That is why seven out of ten residues that interact most frequently with ETH are different from those that interact with the other ligands.

Table 5. 25 distinct top 10 residues (highlighted) for all ligands and their frequencies.

Residue	ETH	NADH	IPF	TCL
ALA21	3,112	7,138	8,414	15,252
ALA190	23,480	3,744	13,714	7,861
ALA197	1,868	14,127	26,114	6,527
ARG42	120	13,959	4,716	1,940
ASP147	22,645	6,795	10,848	9,585
GLY13	3,647	13,479	15,500	19,900
GLY95	5,521	20,288	27,561	23,852
GLY191	22,909	2,162	13,837	839
ILE15	2,079	17,839	13,226	20,397
ILE20	25,480	11,735	23,312	23,393
ILE94	7,570	17,363	26,632	24,460
ILE121	161	15,782	1,430	10,431
ILE193	23,023	6,005	15,519	1,617
LYS164	24,658	14,627	21,821	12,887
MET97	660	14,153	16,661	1,241
MET146	25,368	10,858	18,352	12,625
MET160	21,653	12,355	20,681	6,375
PHE40	446	15,864	4,823	11,220
PHE96	1,355	20,520	19,401	9,292
PHE148	25,961	8,498	15,772	9,923
PRO192	22,816	3,825	13,968	1,240
SER19	3,532	12,619	26,490	23,659
SER93	12,580	12,957	21,726	24,319
SER122	2,421	12,335	19,805	3,111
THR195	17,601	12,348	26,353	20,474

5. CONCLUSIONS AND FUTURE WORK

Considering the explicit flexibility of a receptor molecule in current molecular docking experiments is becoming a common practice. However, this methodology produces huge amounts of data that need to be explored to generate useful information that, in turn, can be used to plan next steps in rational drug design experiments. In our laboratory it was previously performed three different MD simulations of the InhA enzyme receptor: 3.1 ns at 298 K with Amber6, 20 ns at 298 K with Amber9 and 20 ns at 310 K with Amber9. We explored the one with 3,100 ps in molecular docking experiments already performed with four different ligands.

To better understand the role of receptor flexibility in docking experiments, we developed FReDD - a comprehensive repository that stores all MD simulation snapshots and related data about docking results. We performed a set of analyses on FReDD data targeting interesting patterns of receptor-ligand interactions. By exploring this repository for the flexible InhA-ligand interactions, we discovered useful relationships between flexible-receptor residues and the ligands which cannot be assessed using a single, rigid receptor conformation. Such results show how important is the explicit receptor flexibility in molecular docking experiments. It was achieved by the use of a MD simulation trajectory of the receptor to properly represent its flexibility. Having these analyses we can see how effective is centralizing the whole involved data in a proper repository, ease to access and retrieve data. Besides this, FReDD repository allows a transversal analysis by different points of view. As future work, we intend to apply mining algorithms in the stored data aiming at discovering useful patterns among the receptor snapshots. Furthermore, based on the mined results, we intend to perform snapshots selection and check the results in terms of accuracy and precision.

Copy Rights

ACKNOWLEDGEMENT

This work was supported by grants from MCT/CNPq 14/2008 to DDR and in part by grants 410505/2006-4 and 312027/2006-0 to ONS. ONS is a CNPq Research Fellow. ATW is supported by CT-INFO/CNPq 17/2007 PhD scholarship. KSM is supported by a CAPES PhD scholarship.

REFERENCES

Alonso H, Bliznyuk AA and Gready JE, 2006. Combining docking and molecular dynamic simulations in drug design. *Med Res Rev*, Vol. 26, pp 531-568.

Banerjee A et al, 1994. InhA, a gene encoding a target for isoniazid and ethionamide in Mycobacterium tuberculosis. *Science*, Vol. 263, pp 227-230.

Basso LA et al, 1998. Mechanisms of isoniazid resistance in Mycobacterium tuberculosis: enzymatic characterization of enoyl reductase mutants identified in isoniazid-resistant clinical isolates. *J Infect Dis*, Vol. 178, pp 769-775.

Case DA et al, 2006. AMBER 9, University of California, San Francisco.

Cozzini P et al, 2008. Target flexibility: an emerging consideration in drug discovery and design. *J Med Chem*, Vol. 51, pp 6237-6255.

Dessen A et al, 1995. Crystal structure and function of the isoniazid target of mycobacterium tuberculosis. *Science*, Vol. 267, pp 1638-1641.

Gargano F, 2009. The effect of temperature in the complex formed by 2-trans-enoyl-ACP (CoA) redutcase (EC 1.3.1.9) from Mycobacterium tuberculosis and its coenzyme NADH : a molecular dynamics simulation study. PhD Thesis in Celular and Molecular Biology PUCRS, Porto Alegre - Brazil.

Guex N and Peitsch MC, 1997. SWISS-MODEL and the Swiss-PdbViewer: an environment for comparative protein modeling. *Electrophoresis*, Vol. 8, pp 2714-2723.

Irwin JJ and Schoichet B, 2005. ZINC - a free database of commercially available compounds for virtual screening. *J Chem Inf Model*, Vol. 45, pp 177-182.

Kuntz ID, 1992. Structure-based strategies for drug design and discovery. *Science*, Vol. 257, pp 1078-1082.

Kuo MR et al, 2003. Targeting tubercu-losis and malaria through inhibition of enoyl reductase: compound activity and structural data. *J Biol Chem*, Vol. 278, pp 20851-20859.

Lengauer T and Rarey M, 1996. Computational methods for biomolecular docking. *Curr Opin Struct Biol*, Vol. 6, pp 402-406.

Lesk A, 2002. *Introduction to Bioinformatics*. Oxford University Press, New York.

Lybrand TP, 1995. Ligand-protein docking and rational drug design. *Curr Opin Struct Biol*, Vol. 5, pp 224-228.

Machado KS et al, 2007. Automating molecular docking with explicity receptor flexibility using scientific workflows. *Lect Notes Comput Sci*, Vol. 4643, pp 1-11.

Oliveira JS et al, 2004. An inorganic iron complex that inhibits wild-type and an isoniazid-resistant mutant 2-trans-enoyl-ACP (CoA) reductase from Mycobacterium tuberculosis. *Chem Commun*, Vol. 3, pp 312-313.

Oliveira JS et al, 2006. Slowonset inhibition of 2-trans-enoyl-ACP (CoA) reductase from Mycobacterium tuberculosis by an inorganic complex. *Curr Pharm Des*, Vol. 12, pp 2409-2424.

Pearlman DA et al, 1995. AMBER, a computer program for apply-ing molecular mechanics, normal mode analysis, molecular dynamics and free energy calculations to elucidate the structures and energies of molecules. *Comp. Phys. Commun*, Vol. 91, pp 1-41.

Rozwarski DA et al, 1998. Modification of the NADH of the isoniazid target (InhA) from Mycobacterium tuberculosis, *Science*, Vol. 279, pp 98-102.

Schroeder EK et al, 2005. Molecular dynamics simulation studies of the wild-type, I21V and I16T mutants of isoniazid-resistant Mycobacterium tuberculosis enoyl reductase (InhA) in complex with NADH: toward the understanding of NADH-InhA different affinities. *Biophys J*, Vol. 89, pp 876-884.

Totrov M and Abagyan R, 2008. Flexible ligand docking to multiple receptor conformations: a pratical alternative. *Curr Opin Struct Biol*, Vol. 18, pp 178-184.

van Gunsteren WF and Berendsen HJC, 1990. Computer simulation of molecular dynamics methodology, aplications and perspectives in chemistry. *Angew Chem* Vol. 29, pp 992-1023.

Verkhivker GM et al, 2002. Complexity and simplicity of ligand-macromolecule interactions: the energy landscape perspective. *Curr Opin Struct Biol*, Vol. 12, pp 197-203.

Wang F et al, 2007. Mechanism of thioamide drug action against tuberculosis and leprosy. *J Exp Med*, Vol. 204, pp 73-78.

Winck AT et al, 2009. FReDD: supporting mining strategies through a flexible receptor docking database. *Lect Notes Comput Sci*. Vol. 5676, pp 143-146.

Copy Rights

REFINEMENT OF A GENETIC ALGORITHM FOR DOCUMENT CLUSTERING

José Luis Castillo Sequera, José R. Fernández del Castillo Díez and León González Sotos
University of Alcala – Department of Computer Science, Madrid Spain.

ABSTRACT

In this paper we show the strategies used to refine the parameters of a genetic algorithm applied to the field of documentation. Properly assigning these parameters allows us to improve the solution and address the problems of optimization in the evolutionary field successfully. This paper presents an initial introduction on the mentioned topic showing the techniques and strategies implemented in an algorithm designed to cluster documents in a non supervised manner ensuring a balance between diversification or the ability to visit many different regions of the search space, and the intensification or the ability to obtain high quality solutions in these regions. The criteria used for document clustering is based on a fitness function that uses both the similarity and the distance between documents to measure the degree of affinity and closeness between the various documents. We show the strategies used to refine the integration of algorithm parameters and the results obtained by varying the algorithm parameters to improve performance and obtain an acceptable document cluster at the end of the evolution, and provide at least two possible groups among all documents, placing documents by affinity. The proposal to be presented as an alternative to traditional methods in Information Retrieval.

KEYWORDS

Data Mining, Genetic Algorithm, Information Retrieval, Documentation, Optimization Methods

Copy Rights

1. INTRODUCTION

Both in industry and science there are some real problems regarding the optimization of difficult solution characterized by computational complexity, because the available exact algorithms are inefficient or simply impossible to implement. The metaheuristics (MHs), [2] are a family of approximate methods of general purpose consisting in iterative procedures that guide heuristics, intelligently combining different concepts to explore and exploit properly the search space. Therefore, there are two important factors when designing MHs : intensification and diversification [7]. The diversification generally refers to the ability to visit many different regions of search space, while intensification refers to the ability to obtain high quality solutions in these regions. A search algorithm must achieve a balance between these two factors so as to successfully solve the problem addressed. On the other hand, Information Retrieval (IR) can be defined as the problem of information selection through a storage mechanism in response to user queries [3]. The Information Retrieval Systems (IRS) are a class of information systems that deal with databases composed of documents, and process user's queries by allowing access to relevant information in an appropriate time interval. Theoreticly, a document is a set of textual data, but technological development has led to the proliferation of multimedia documents. It is clear that IR can not be carried out intelligently unless the IRS use intelligent systems that mimic the natural evolution process.

Genetic Algorithms (GAs) [2] are inspired by MHs in the genetic processes of natural organisms and in the principles of natural evolution of populations. The basic idea is to maintain a population of chromosomes, which represent candidate solutions to a specific problem , that evolve over time through a process of competition and controlled variation. One of the most important components of GAs is the crossover operator. Considering all GA must have a balance between intensification and diversification that is capable of augmenting the search for the optimal, the crossover operator is often regarded as a key piece to improve the intensification of a local optimum. Besides, through the evolutionary process, every so often there are species that have undergone a change (mutation) of chromosome, due to certain evolution factors, as the mutation operator is a key factor in ensuring that diversification, and finding all the optimum feasible regions. The appeal of GAs as search procedures has led to the design of specific GAs, but they all must provide and ensure a balance between diversification and intensification to find the optimum, because their performance depends on proper parameter selection of population size, genetic operators, rate of probability, selection scheme, etc.

Efficiently assigning GA parameters optimizes both the quality of the solutions and the resources required by the algorithm. This way, we can obtain a powerful search algorithm and domain independent, which may be applied to a wide range of learning tasks [7]. One of the many possible applications to the field of IR might be solving a basic problem faced by an IRS: the need to find the groups that best describe the documents, and allow each other to place all documents by affinity. The problem that arises is in the difficulty of finding the group that best describes a document [4], since they do not address a single issue, and even if they did, the manner the topic is approached can also make it suitable for another group. Therefore, this task is complex and even subjective as two people could easily assign the same document to different groups using valid criteria. This paper presents a GA applied to the field of documentation, the algorithm improved itself by refining its parameters, offering a balance between intensification and diversity that ensures an acceptable optimal fitness along an unsupervised document cluster. The main contribution is in the way documents are being represented, all of its parameters that ensure a balance between diversity and intensification of feasible regions, and the use of metrics that evaluate the fitness of GA with these parameters.

2. DOCUMENTARY BASE

In this study we make use of two collections, the "Reuters 21578" collection and a Spanish documentary base that includes editorials of "El Mundo" from 2006 and 2007 in an open acces format.

Reuters Documentary Base consists of real news wires that appeared in Reuters in 1987, this collection is becoming a standard within the domain of the automatic categorization of documents and is used by many authors in this area [3] [4]. The collection consists of 21578 documents distributed in 22 files. We developed a documentary process named NZIPF [6] to generate documentary vectors that feed the system.

The documentary process consists of several stages of document processing, each of which represents a process that was developed on the base document to obtain documentary vectors more efficiently. The process is outlined in Figure 1.

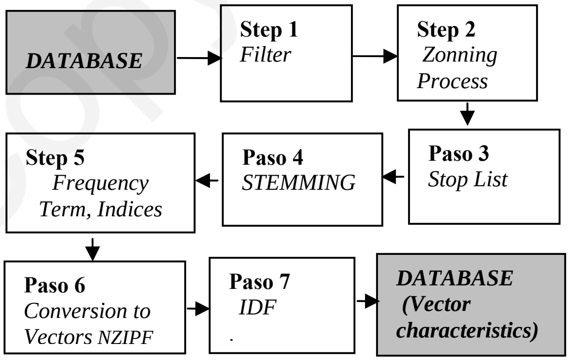


Figure 1. Documentary process conducted

Copyrights

3. EXPERIMENTAL ENVIRONMENT

Within the testing environment there should be a user to provide documents that are meant to be grouped. The role of the user who provides documents will be represented by the samples of "very few (20), few (50), many (80) and enough (150)" documents, with the requirement that belonged to only two categories of Reuters or distribution of Editorials in Spanish represented by their feature vectors stemmer. Figure 2 shows the documentary environment that we used for the experiments, it is important to note that, unlike the algorithms of the type monitored, where the number obtained groups needs to be known, our algorithm will evolve to find the most appropriate structure, forming the groups by itself in an unsupervised way.

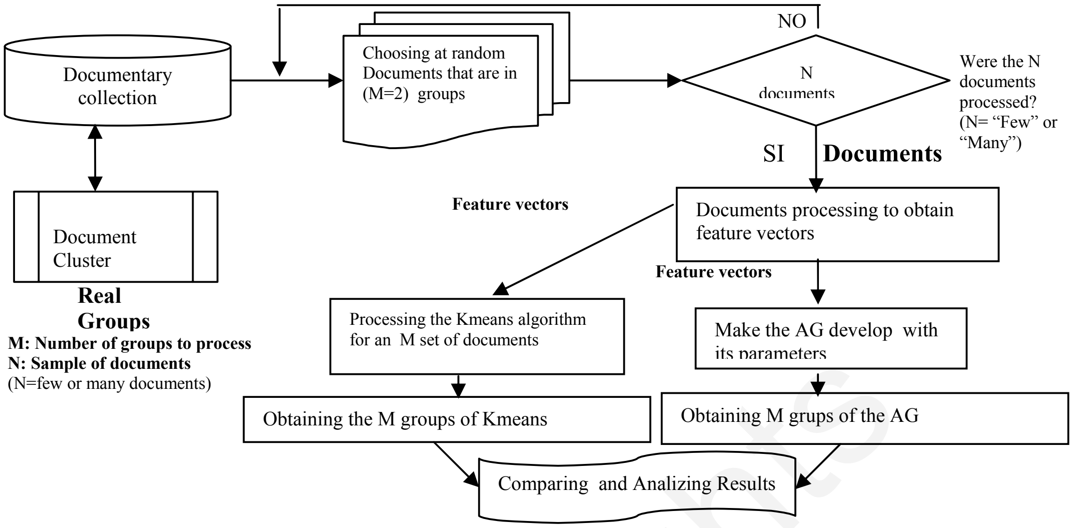

NO
Documentary collection
Choosing at random Documents that are in (M=2) groups
N documents
Were the N documents processed? (N= "Few" or "Many")
SI
Documents
Feature vectors
Documents processing to obtain feature vectors
Feature vectors
Document Cluster
Real Groups
M: Number of groups to process
N: Sample of documents
(N=few or many documents)
Processing the Kmeans algorithm for an M set of documents
Make the AG develop with its parameters
Obtaining the M groups of Kmeans
Obtaining M grups of the AG
Comparing and Analizing Results

Figure 2. Experimental environment used in the tests with the GA.

Due to the nature of simulation of GA, its evolution is pseudo-random, this translates into the need for multiple runs with different seeds to reach the optimal solution. The generation of the seed is carried out according to the time of the system. For this reason, the experiments with GA were made by carrying out five executions to each of the samples taken from experimental collections. The result of the experiment will be the best fitness obtained and their convergence. To measure the quality of the algorithm, the best solution obtained and the average of five runs of the GA must be analized.

Within the experiments with our experimental environment, we used samples of documents "very few (20), few (50), many (80) and enough (150)" documents with the requirement that they belonged only to two categories of Reuters collections or Editorials. Each of the samples processed with five different seeds, and each of the results are compared with the method *"Kmeans."* Then, each experiment was repeated by varying the rate of probability of genetic algorithm operators, using all the parameters shown in table 1.

Copy Rights

4. GENETIC ALGORITHM FOR DOCUMENT CLUSTERING

4.1 Individuals

The population consists of a set of individuals, where each of it is made of a linear chromosome that is represented through a tree structure (hierarchical structure). An individual shall formed on a binary tree structure *cluster all documents* prepared at the top, where each document consists of a *feature* ***vector.*** The vector will consist of the weighted values of the frequencies of the stemmer terms that have been selected to implement the document processing scheme [6]. This representation will be attempted to evolve so that the chromosome will undergo genetic changes and find the groups "Clusters" more appropriate for all documents of the IRS. Within the root node we will have our fitness function *(fitness)* that measure the quality of the resulting clustering. Depending on the number of documents that need to be processed and the depth (height) of the tree you want to create, chromosome may be of variable length. Figure 3 shows the initial generation (0). A scheme of tree-based representation is adopted in order to allow the encoding of sufficiently complex logical structures within a chromosome. The search area for the GA is the space of all possible trees that can be generated, resulting from the whole relevant functions and terminals. This way we can evolve individuals of various shapes and sizes, allowing evolution to decide what are the best settings [5].

Although the initial population is random, there is a defined set of parameters governing the establishment of such individuals. For example, *there should not be created in the initial set two equal individuals,* for this production rules are created to ensure the compliance with this condition. The above mentioned rules require that the building grammar of each individual nodes takes place in Preorder.

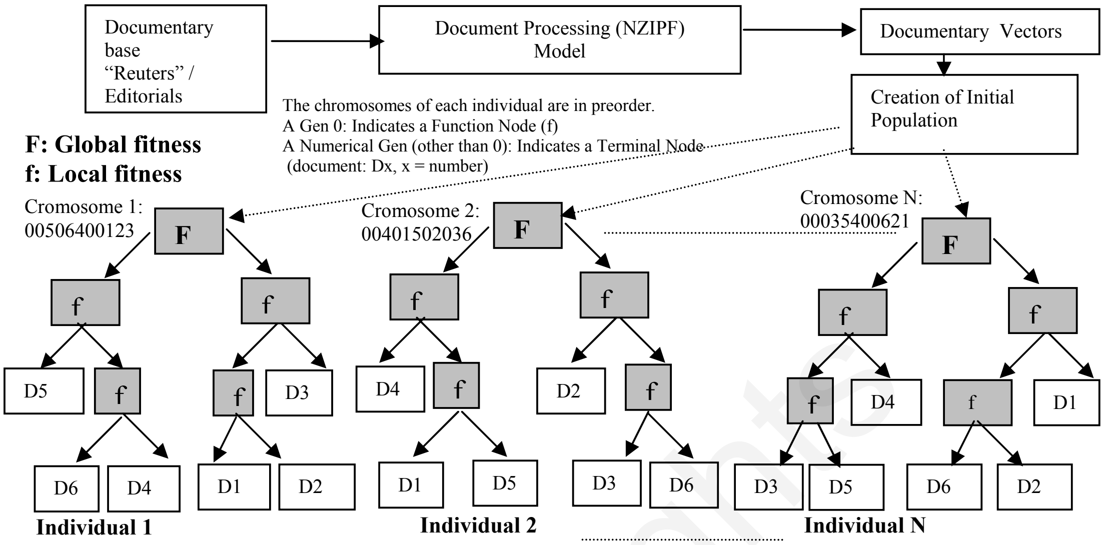

Documentary base "Reuters" / Editorials
Document Processing (NZIPF) Model
Documentary Vectors
Creation of Initial Population
The chromosomes of each individual are in preorder.
A Gen 0: Indicates a Function Node (f)
A Numerical Gen (other than 0): Indicates a Terminal Node (document: Dx, x = number)
F: Global fitness
f: Local fitness
Cromosome 1: 00506400123
Cromosome 2: 00401502036
Cromosome N: 00035400621
F
f
D5
D6
D4
D1
D2
D3
Individual 1
Individual 2
Individual N

Figure 3. Initial Population of Individuals GA (generation "0")

4.2 Production Operators

The production operators are applied to each new generation. One or two individuals can be taken to produce new individuals for next generation by applying the transformations imposed by the operator. Both mutation operators and crossover will be implemented indistinctably. Both operators depend on a mutation probability and / or cross that is assigned to GA. A *mutation operator* is applied on nodes (documents), selecting an individual from the population using the tournament method, and then randomly select a pair of terminal nodes of that individual to mutate its terminal nodes, generating a new individual transposing the nodes that have been chosen (Figure 4). This same process is appliable for subtrees [5]. but we only incorporate it in the population if the new individual has a better fitness.

For the crossover operator, an operator based on *mask* ***crossover*** is applied [5], which selects through tournament method two parent individuals, randomly chooses the chromosome of one parent to be used as *"crossover mask of the selected individual¨.* The crossing is done by analyzing the chromosome of both parents. If both chromosomes have at least one function node (node 0), the chosed father mask is placed, but if we find documents in the chromosomes of both parents, then, the father *"not elected"* document will be selected and we'll use it as pivot on the father *"elected"* (mask) to make the crossing that corresponds to the mentioned father, while interchanging the chromosomes of the mentioned father. This *creates a new individual,* and ensure that in the given chromosome set there are the same structural characteristics of the parents (having documents not repeated), but we only incorporate it in the population if the child has a better fitness than their parents. For example, if we have five documents and the following parent chromosomes selected 0 0 2 1 0 5 0 3 4 0 0 0 5 3 0 2 1 4. The created chromosome after applying the crossover operator proposed would be 0 0 0 1 3 0 2 5 4 (Figure 5).

In either case, new individuals created after the mutation or the crossing, are incorporated into the new population only if they improve the fitness of the original individual (or his parents).

Copy Rights

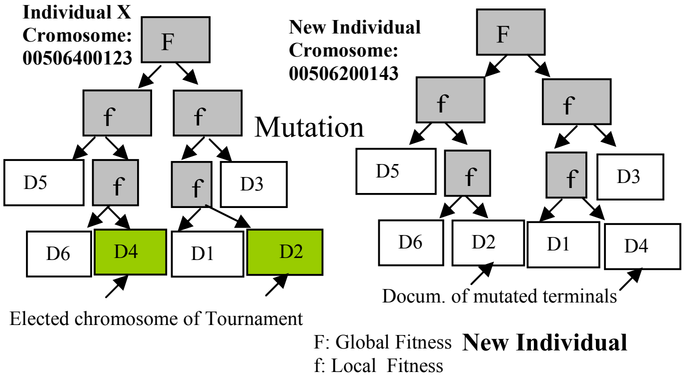


Figure 4. Basic mutation operator applied to terminal

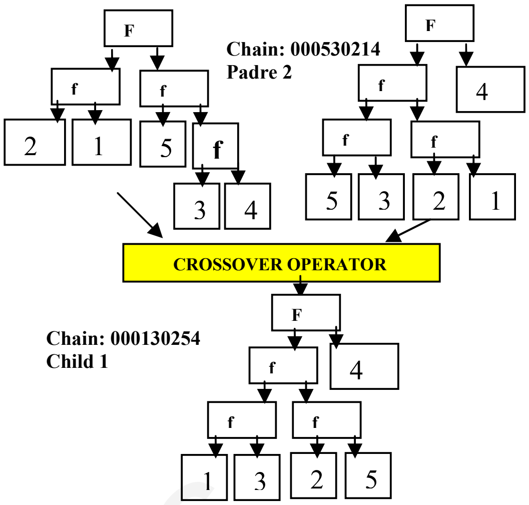


Figure 5. Crossover operator (crossover mask)

5. PARAMETER CONTROL

For its size, and the influence that small changes have on the behavior of the GA during the experiments, the choice of parameter values that are going to be used appears as a critical factor . For their election we paid attention to the variation of the GA performance indicators when it changed the value of any of these, specifically the evolution of the successes and the evolution of *"fitness"*. Therefore, these parameters are very important parts as they directly influence the performance of the GA. These parameters can be treated independently, but the overall performance of the algorithm does not depend exclusively on a single parameter but on a combination of all parameters. The problem of correctly setting the parameters allows the classifying of GA [7] in two cathegories: The configuration of the parameters before execution, which is called *tune-up parameters* (technique for finding the optimal parameters of GA before running the algorithm, and then set the optimal algorithm to these values), unfortunately this method should be made for each GA that is implemented, which hampers its usefulness, and the *control parameters* based on changing the values of GA parameters during execution, seeking an optimum value for the algorithm. Because GA have several **parameters** that must be carefully chosen to obtain a good performance and avoid premature convergence, in our case and *after much testing,* we opted for the control of parameters, and some strategies such as:

To control the ***population size*** we use the strategy called GAVAPS (Genetic variation in population size) proposed by Michalewicz [7], using the concept of age and lifetime. When creating the first generation all individuals are assigned a zero age, referring to the birth of the individual, and every time a new generation is born the age of each individual increases by one. At the same time an individual is born it is assigned a lifetime, which represents how long it will live within. Therefore, the individual will die when it will reach the given age. The lifetime of each individual depends on the value of its fitness compared to the average of the entire population. Thus, if an individual has better fitness will have more time to live, giving it greater ability to generate new individuals with their features. In our case, we allow each generation to generate new individuals with similar characteristics with this strategy. Therefore, we adopt this approach essentially the best individuals from each generation, and apply it to maintain ***elitism*** in the following generations, thus ensuring optimum intensification of available space, while keeping them during their lifetime. However, to ensure diversity we *randomly* generate *the remaining individuals* in each generation. This way, we explore many different regions of the search space and allow for balance between intensification and diversity of feasible regions. In all cases, the population size has been set at 50 individuals for the experiments conducted with samples following the suggestion of [1], which advises working with a population size between *l* and *2l* in most practical applications (the length of chromosome l) In our case, "l" the length of our chromosome is always equal to: ***2 * number of documents to cluster -1.***

Copyrights

5.1 Tests to determine the Value of the Rate of Mutation Operator and Crossover Operator Rate

We began conducting an analysis of system behavior by varying the rate of mutation operator in a wide range of values to cover all possible situations. Thus, for the rate of mutation operator discussed a wide range of values in the range of: 0.01, 0.03, 0.05, 0.07, 0.1, 0.3, 0, 5, 0.7; that allowed us to apply the mutation operator of GA in different circumstances and study their behavior. For the study to determine the optimal value of the rate of crossover operator, is traced the interval from 0.70 to 0.95; value high, but oriented to frequently apply the operator we designed, and following Schaffer's recommendations[8] which states that an optimum value for the ***mutation probability*** is much more important than the crossover probability, and choose to make a more detailed study of the odds ratio in our experiments. As a quality index value of the operator was given to the number of hits of the GA. In the following graphs, we show the average number of hits returned by the GA for samples of 20, 80 and 150 documents, changing the mutation rate, and show the hit factor of the GA against the mutation rate As for the ***size of the tournament,*** the value 2 has been chosen, because the binary tournament has shown a very good performance in a large number of applications of EAs [2] [7]. Although determining a optimal *fitness* function is not one of the fundamental objectives of this experiment, we have tried to add in a single value the mesuring results as powerful and distinct as are the Euclidean distance and the Pearson correlation coefficient, as illustrated in the following equation:

$$\textbf{Fitness} = \textbf{Min} (\alpha\ \textbf{Distance} (\textbf{Documents}_i) + (1\text{-}\alpha) (1/\ \textbf{Similarity} (\textbf{Documents}_i)))$$

where: α: would be the parameter that adjusts the distance
1- α would be the parameter that adjusts the similarity

Therefore, to find and the adjustment coefficient α that governs the weight that is to be given to both the distance as the inverse of similarity of the cluster documents, we've made many parameter controlled tests in order to obtain a value that allows an adequate contribution of both metrics with respect to fitness., finally finding a value for of 0.85. The **number of maximum generations** the system has been set to is 5000, but this parameter may vary depending on the convergence of the algorithm. As for the **number of stemmer terms** to be used for representing the feature vectors of each of the documents have been selected through the NZIPF processing method [6]. Finally, we have established a limit called the **threshold of depth** for individuals (trees). Such a threshold, in the case of *"very few and few documents"* take the value of 7, and for the*"many and enough documents"* is set 10, so that the individual generated by the GA does not have an exaggerated depth compared to the number of documents to be processed.

5.2 Studies to determine the Value of α in the GA

We use the distribution Reuters 21 of be that greater dispersion across your documents and apply the GA varying the value of α in each of the tests with the usual parameters, always trying to test the effectiveness of the GA. We analyzed the relationship between fitness and the value of α (figure 6) using the values in table 1.

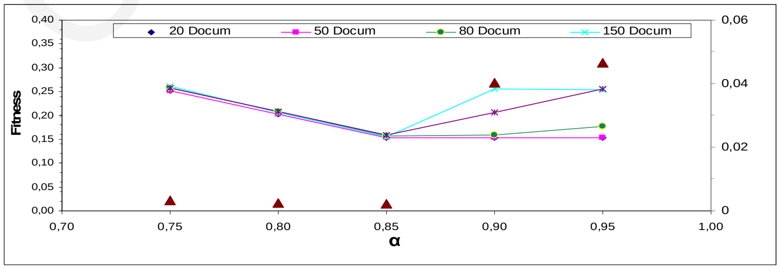


Figure 6. Best Fitness versus α values for different samples of documents of the Reuters Collection: Distribution 21**.**

In figure 6, we can see that there is an increased dispersion of fitness values over 0.85, due to the increased contribution of Euclidean distance which makes it insensitive to fitness to find the clusters. The results, suggest that a value of α close to 0.85, provides better results because it gives us more effective in terms of number of hits, and a better fitness of the algorithm. This was corroborates with other distribution.

Table 1. Parameters taken into consideration for the Evolutionary system

Parameters	Values
Population size (tree number)	50
Número de evaluaciones (Generaciones)	5000 maximum
Tournament size	2
Mutation Probability (Pm)	0.01, 0.03, 0.05, 0.07, 0.1, 0.3, 0, 5, 0.7
Crossover Probability (Pc)	0.70,0.75,0.80,0.85,0.90,0.95
Document cuantity	Very Few, Few, Many, enough
α coeficients	0.85
Depth Threshold	7 /10

6. EXPERIMENTAL RESULTS

To analyze the results, and to verify their effectiveness, we compared the results of the GA with the existing real groups of the document collection, and also compared the results with another supervised type of clustering algorithm in optimal conditions (Kmeans) We analized the following:

a) **Cluster efectiveness:** It is the most important indicator of the comparison of results considering the quality of the cluster. An analyzing process was carried out to see the successes achieved with the best fitness of GA, and also the average scores in all executions of the GA.

b) **Fitness evolution.** Analysis was carried out to see the evolving fitness in each of the performances, assessing their behaviour and successes of the GA when varying the probability rate.

c) **Convergence of the algorithm**: In which process the GA obtains the best fitness (best cluster).

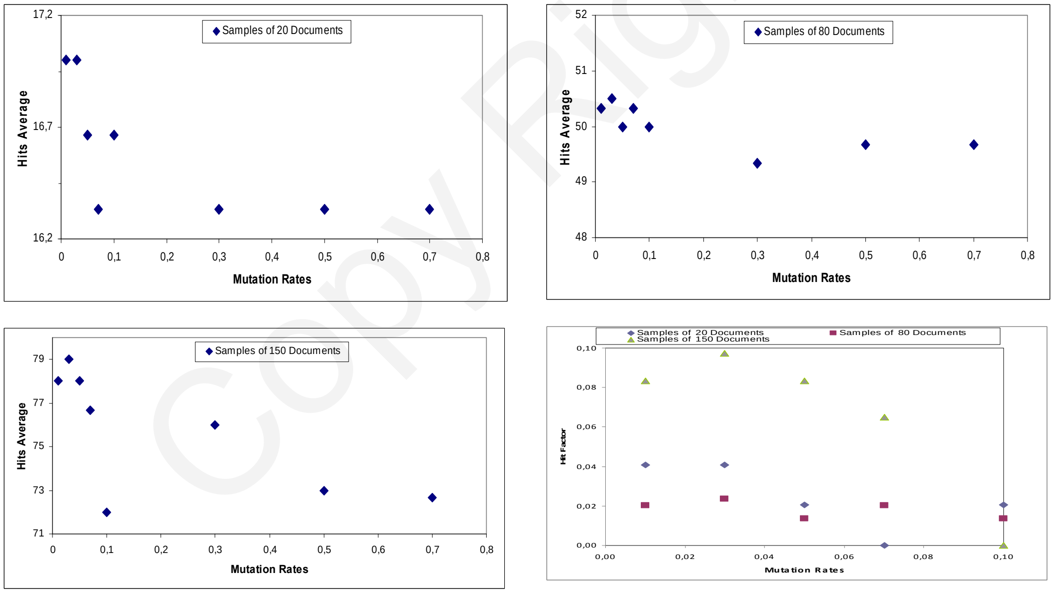


Figure 7. Hits average of GA with samples 20, 80 and 150 documents varying mutation rate and hit the GA.

Since, the GA parameters directly affect the fitness behavior, before the experiments, we performed a comprehensive analysis of all GA performances, in order to determine its robustness and adjusting each of its parameters. Finally, we experimentally used the parameters discussed in Table 1 and analyzed the behavior of the algorithm. We show in Figure 7 the average number of hits returned by the GA for samples of 20, 80 and 150 documents, changing the mutation rate, and show the hit factor of the GA against the mutation rate. We appreciate that we got the best performance with a rate of 0.03, this result shows that the best medium fitness could also be obtained by using this rate. We corroborated that conduct with another collection.

In addition, we analyzed the incidence of crossover operator on the final results. The figures 8 show the behavior of the crossover rate versus hits average with very few samples (20), many (80) and many

Copy Rights

documents (150) respectively. Besides a comparative analysis is the success factor of GA varying the crossover rate. It makes clear, the GA performed better when using a rate of 0.80 for the crossover operator, regardless of the sample. Therefore, this value appears to be ideal if we maximize the efficiency of the algorithm, which is why we conclude that is the rate that gives us better results.

To corroborate the results of the GA, we compare their results with the Kmeans algorithm , which was processed with ***the same samples,*** passing as input the number of groups that needed to be obtained.

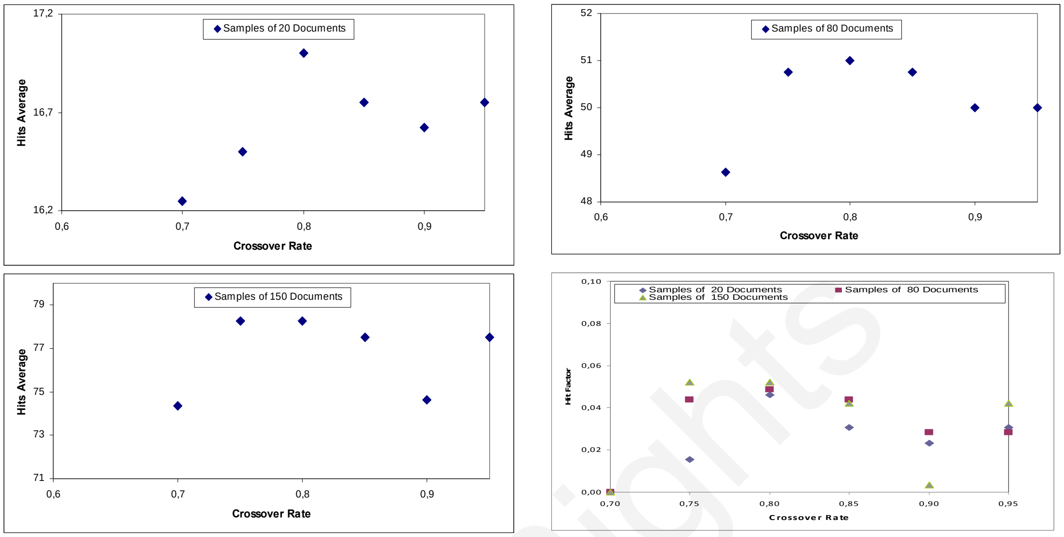


Figure 8. Hits average of GA with samples 20, 80 and 150 documents varying crossover rate and hit the GA**.**

7. CONCLUSION

Because GA have been already used in IRS, considering the results we can conclude that the GA had a favourable evolution, offering optimal document cluster in an acceptable and robust manner, based on a proper adjust of the parameters. We proved that ***the medium effectiveness*** of the GA is very acceptable, being in most cases better than Kmeans supervised algorithm, but with the added advantage that we processed the documents in an unsupervised way, allowing evolution perform clustering with our adjustment.

ACKNOWLEDGMENTS

The author should like to express their appreciation to Miss Cristina Huidiu for his continuous support.

REFERENCES

1. [Alander, 1992] Alander. J*"On optimal populations size of genetic algorithms"* Proc CompEuro 1992.
2. [Bäck, 1996] Bäck T, *"Evolutionary Algorithms in theory and Practice"*, Oxford University Press, 1996.
3. [Berry Michael, 2004] M. Berry , Survey of Text Mining – Clustering and Retrieval, Springer 2004.
4. [Berry Michael, et al,2008] M. Berry, Malu Castellano Editors:"*Survey of Text Mining II*", Springer,2008.
5. [Castillo,Fernandéz,León,2008] "*Information Retrieval with Cluter Genetic*" IADIS Data Mining, 2008.
6. [Castillo,Fernandéz,León,2009] "*Feature Reduction for Clustering with NZIPF*" IADIS e-Society 2009.
7. [Michalewicz, 1999] Michalewicz Z.*"Genetic Algorithms + Data Structures = Evolution"*. Springer-1999.
8. [Schaffer et al,1989] Shaffer, et al, 1989 "*A study of control parameters performance on GA for function optimization"*

Copy Rights

FACE RETRIEVAL USING SALIENT FACIAL POINTS

Luigi Cinque and Enver Sangineto
Computer Science Department, University of Rome “Sapienza”,
Via Salaria 113, 00198, Rome, Italy

ABSTRACT

We present in this article a system for face retrieval based on the relative positions of a set of prefixed anatomically salient facial features (*landmarks*). Landmarks are extracted from the face image automatically using a two stage approach. In the first phase, a local classification of SIFT descriptors provides a set of landmark candidates. In the second phase, a modified version of the Hausdorff distance is proposed to correct possible errors of the local phase using global shape information.
Once the landmark points are extracted, a face image is represented by means of the facial feature positions. A nearest-neighbor approach is used in order to retrieve from a database of suspect individuals those faces whose landmark dispositions best aligns with the landmarks of the input query. It is important to note that landmark selection can be done with every kind of face representation: either drawings or real images are dealt with at the same manner. For this reason, our system is able to compare real images with drawings while common face recognition systems cannot.

KEYWORDS

Face Recognition, Facial Landmark Localization, Biometry.

1. INTRODUCTION

We present in this paper a computer-based system for real-time comparison with a face database of suspects for retrieval purposes. The positions of a set of landmark points on the query face image is used by the system to search for similar faces in a database of possible suspects. This is the main novelty of our approach with respect to existing face recognition systems. Indeed, common face recognition approaches can either compare a real photo (e.g., the user's query showing the suspect) with a database of real photos or compare a query drawing with a drawing database. As far as we know, no existing face recognition system is able to compare drawings (i.e. the common product of a human-made identikit) with real photos.

A lot of computer-based identikit composition systems have been proposed proposed [Bruce, Gillenson, Laughery, Pentland, Brunelli 1996, Shepherd]. Most of them however, either have limited drawing capabilities or do not provide sufficiently robust retrieval instruments (or do not provide these last at all).

In SpotIt [Brunelli 1996], for instance, the identikit construction is based on a set of user settable sliders which define suitable interpolation values for a set of different face features. For each main face feature (nose, mouth, eyes, etc.), the system relies on a set of training data. Suppose that $T_j = \{ F_i^j \}_{i=1, ..., N}$ is a set of image samples of feature F_j (e.g., a set of real images, each image showing the nose of a given person). T_j is the training data for the j-th feature. The system then performs a standard Principal Component Analysis (PCA) to compress the image dimensions. PCA is based on the extraction on the most relevant statistical information from a training data and on representing the training elements themselves in a new space whose dimensions are the eigenvectors of the covariance matrix of the set $\{ F_i^j - MF^j \}_{i=1, ..., N}$, being MF^j the mean value of T_j. Suppose that $\{ U_k^j \}_{k=1, ..., m}$ is the set of eigenvectors selected for the j-th feature. Then each element F_i^j in T_j can be represented as:

$$F_i^j = \overline{F^j} + \sum_{k-1}^{m} c_k^j U_k^j \qquad (1.1)$$

and the values of the coefficients c_k^j ($k=1, ..., m$) are all we need for representing F_i^j. At run-time SpotIt provides the user with the possibility to manually set the coefficients c_k^j for each j-th face feature using a interactive menu composed of moving sliders. The values $c_1^j, ..., c_m^j$, for each j, are finally used by the system for both the

Copy Rights

synthesis of a new face in the identikit formation phase and in retrieving those real face images stored in the suspect database which have similar values of the coefficients.

Even if this system is sufficiently complete providing both composition and retrieval capabilities, there are some limitations in both phases. First, feature synthesis using coefficients variability in a prefixed PCA-based feature space does not provide a full degree of freedom in identikit composition. For instance, the user cannot define the feature shape: even if a minor shape deformation is possible, e.g., moving up and down the position of the nose or changing its dimensions, the complete autonomy available for a human drawer when she/he draws using her/his hands cannot be completely simulated. Moreover, the subspace obtained with PCA-based compression techniques usually allows to accurately represent the images *of the training database* without significant information loss but it degrades when used to represent images *of persons/classes of objects do not belonging to the training set*. For this reason, the generality with which an identikit can be built and used for retrieval purposes is limited. Finally, PCA-based retrieval systems assumes a parametric statistical distribution of the feature values, which is not always true.

Vice versa, our proposed system is based on the following characteristics.

- *Retrieval accuracy*. Our system represents a face image using a vector of *landmark positions* which is independent of the image type, since landmarks can be localized in either real images and drawings. The landmark position vector is subsequently used for retrieval purposes (Section 3). Differently than in SpotIt [Brunelli 1996], we do not rely on PCA techniques. This makes it possible to add/remove individuals from the system's database without the need of re-training the system. The retrieval process is performed using a *Nearest Neighbor* approach based on the landmark representation. No parametric or other statistic distribution assumption of the feature values is done.
- *Real time performance*. Even if we do not perform any data compression, space and time requirements are however satisfied by indexing the system's database using a *k-d tree* data organization [Del Bimbo] and a low dimensional landmark vector.

The rest of the paper is organized as follows. In Section 2 we present how landmarks are *automatically* localized in a given face. In Section 3 we show how the landmark position vector is used in order to retrieve those individuals of the system's database which are the most similar to the input query. In Section 4 we show some experimental results of this approach. Finally, we conclude in Section 5.

2. LANDMARK LOCALIZATION

After a standard preprocessing step (we convert the input image to an 8-bit grayscale image, apply the Viola Jones Face [Viola], and resize the face area returned by the face detector to fixed dimensions), we extract the SIFT (Scale Invariant Feature Transform) features as reported in [Lowe]. Then, SIFT descriptors are analyzed in order to localize facial landmarks.

The facial landmark localization process is based on a combination of local and global information. At training time, we use a training set of *T* images (*T = 400*) in which the coordinates of each of 20 fixed landmarks are manually annotated. The 20 landmark adopted in this work are the same used in [Cristinacce] and in the BioID dataset [Jesorsky] (see Figure 1).

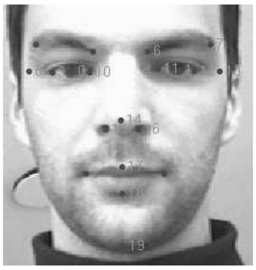


Figure 1. The 20 facial landmark positions as defined in the BioID dataset [Jesorsky].

From these *T* images we extract SIFT descriptors from stable points and we create 21 training classes S_0, ..., S_{20}: one class for each landmark plus a "non class" (S_{20}) representing areas of the face far from every landmark. S_i ($0 <= i <= 19$) is composed of all those SIFT descriptors extracted from the *T* training images whose interest

Copy Rights

point (i.e., the point in which the descriptor has been extracted) is close to the i-th landmark. S_{20} contains all the remaining descriptors not included in any other class. We use classes $S_0, ..., S_{20}$ to train a K-Nearest Neighbor (K-NN) based classificator.

Moreover, an average landmark displacement A is computed using the T coordinates of all the annotated points in the T images. $A = \{ a_1, ..., a_{20} \}$ is a set of 20 point coordinates, where $a_i = (x_i, y_i)^T$ (0 <= i <= 19) is the average position of the i-th landmark with respect to the window returned by the face detector and a_i is estimated using the training set.

At detection time, the SIFT descriptors extracted from the input image are classified using the K-NN (local) classificator. The result of this local classification step is a set of candidate landmark positions $C = \{ C_0, ..., C_{19} \}$. In turn, $C_i = \{ c_{i1}, ..., c_{ih} \}$ ($0 <= i <= 19$) is the set of candidate positions for the i-th landmark. Note that, due to possible false positives and false negatives occurring in the local classification step, we can have that $i_h > 1$ or that C_i is the empty set.

Possible errors (i.e., possible false positives and false negatives) in C are corrected using geometric global information: we use a modified version of the Housdorff distance in order to compute the position (translation offset) of A which minimizes the mismatch with C. Below we provide details on both the “standard” Housdorff distance and our proposed modified version.

Given two finite sets of 2D points $Z = \{ z_1, ..., z_p \}$ and $W = \{ w_1, ..., w_q \}$ (where both z_i, $1 <= i <= p$ and w_j, $1 <= j <= q$ are bidimensional points), the *direct* Housdorff distance is defined as:

$$h(Z,W) = \max_{z \in Z} \min_{w \in W} ||z - w||, \qquad (2.1)$$

where || . || is some predefined norm (e.g., the Euclidean norm). The function $h(Z,W)$ identifies a point z' in Z which is the farthest from any point of W and measures the distance from z' to its nearest neighbor in W. If $h(Z,W) = d$, then each point of Z is within distance d of some point of W, and there exist at least one point of Z (z') which is exactly distance d from the nearest point of W (z' is the most mismatched point). Eq. (2.1) can be easily extended to deal with translations of one of the two involved sets:

$$M_T(Z,W) = \min_{t \in T} h(Z \oplus t, W), \qquad (2.2)$$

where $\oplus$ is the Minkowski notation: $Z \oplus t = \{ z + t \mid z \text{ in } Z \}$, $t = (t_x, t_y)^T$ is a translation offset and T is the finite set of possible translation vectors.

Copy Rights

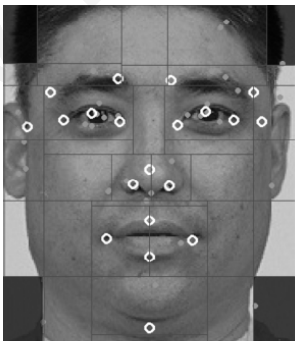


Figure 2. An example of face landmark detection. Green dots represent the SIFT features extracted from interest points, white circles represent the 20 facial landmarks automatically localized by the system.

We are now interested in computing the mismatch between our pre-built set of average landmark positions A and the landmark candidates resulting from the local classification phase (C), taking into account that the i-th landmark average position a_i can be compared only with the elements C_i of C. For this reason, we define the *typized* direct Housdorff distance $l(A,C)$ as follows.

$$l(A,C) = \max_{a_i \in A, i \in J} \min_{c \in C_i} ||a_i - c||, \qquad (2.3)$$

where $J = \{i_1, i_2, ... \}$ is the the set of indexes corresponding to those candidate sets Ci_1, Ci_2, ... which are non-empty. Combining Eq.s (2.2) and (2.3) we obtain:

$$L_T(A,C) = \min_{t \in T} l(A \oplus t, C). \qquad (2.4)$$

In practice, we iterate for a small number of translation offsets selecting the value of t' which minimizes $l(A$

$\oplus$ *t, C). A* $\oplus$ *t'* is finally regarded as the most likely displacement of our set of 20 landmarks in the input image (e.g., see Figure 2).

In Section 3 we show how such a representation can be easily and efficiently used for retrieval purposes. Notice that the landmark selection is independent of the query image type and the landmark positions do not change with the age. As a consequence, drawings as well as real images, with or without beard, glasses, etc., can be dealt with in the same manner. The recognition process accuracy needs only either an identikit or a photo in which the general anatomy of the searched individual is represented with a sufficient approximation.

3. FACE RETRIEVAL

Face retrieval is performed using only *anatomic* information defined by the *landmark vector* $x = A \oplus t'$ (see Section 2). Such information is independent of aging effects, lighting conditions and identikit type (either real or drawing images bring to the same landmark vector extraction). Let L_Q be the landmark vector extracted from the query image identikit Q. Using a nearest neighbor approach the system is able to output an ordered list of the images of the database ranked using their similarity with respect to Q.

More in details, for each image I of the system's database we compute the corresponding landmark vector L_I at the moment in which I is added to the repository. At run time, given a database of N images $I_1, ..., I_n$, we compare L_Q with all the corresponding image landmark vectors $L_{I1}, ... L_{In}$. Nevertheless, before comparing L_{Ii} ($1 <= i <= N$) with L_Q, the two vectors need to be aligned, i.e., we need to compensate for possible scale, (in-plane) rotation and translation differences between the original images I_i and Q with respect to which L_{Ii} and L_Q are represented. For this reason, both L_{Ii} and L_Q are normalized with respect to a reference system which is *symmetric transformation independent* (i.e., it is independent of possible rigid symmetric transformations such as translations, rotations and scale changes).

In order to deal with translations of the I's original reference system with respect to the Q's one, Landmark 0 (the nose tip) is used as the center of a new reference system and all the other landmark points are computed as offset with respect to l_0. If:

$$L = (x_0, y_0, x_1, y_1, ..., x_i, y_i, ...x_{k-1}, y_{k-1})^T$$

is a landmark vector computed with respect to a given image I, its translation independent representation is a $2(k-1)$ dimensional vector defined as:

$$L^{(t)} = (x_1 - x_0, y_1 - y_0, x_2 - x_0, y_2 - y_0, ..., x_i - x_0, y_i - y_0, ...x_{k-1} - x_0, y_{k-1} - y_0)^T. \qquad (3.1)$$

Scale independence is obtained using the translation-normalized landmarks $l_7^{(t)}$ and $l_{19}^{(t)}$ which define the height of the face (see Figure 1). Let $d = y_7^{(t)} - y_{19}^{(t)}$. The scale-normalized representation of the landmark vector is:

$$L^{(s)} = (x_1^{(t)}/d, y_1^{(t)}/d, ..., x_i^{(t)}/d, y_i^{(t)}/d, ...x_{k-1}^{(t)}/d, y_{k-1}^{(t)}/d)^T. \qquad (3.2)$$

Finally, in order to reduce possible slopes of the head with respect to the shoulder, we define a face symmetry axes a as the line passing through the landmarks l_0 and l_{19} (see Figure 3) and we normalize all the landmark points with respect to the angle *theta* formed by the intersection of a with the horizontal axes. Hence, the final rotation-normalized landmark vector is given by:

$$L^{(r)} = (x_1^{(r)}, y_1^{(r)}, ..., x_i^{(r)}, y_i^{(r)}, ...x_{k-1}^{(r)}, y_{k-1}^{(r)})^T, \qquad (3.3)$$

being:

$$l_i^{(r)} = (x_i^{(r)}, y_i^{(r)})^T = (x_i^{(s)} \cos\theta - y_i^{(s)} \sin\theta, x_i^{(s)} \sin\theta + y_i^{(s)} \cos\theta)^T. \qquad (3.4)$$

If:

$$L_Q^{(r)} = (x_1^{(r)}, y_1^{(r)}, ..., x_i^{(r)}, y_i^{(r)}, ...x_{k-1}^{(r)}, y_{k-1}^{(r)})^T$$

is the normalized query landmark vector and:

$$L_I^{(r)} = (\overline{x}_1^{(r)}, \overline{y}_1^{(r)}, ..., \overline{x}_i^{(r)}, \overline{y}_i^{(r)}, ...\overline{x}_{k-1}^{(r)}, \overline{y}_{k-1}^{(r)})^T$$

is the normalized landmark vector of a generic image I of the system's database, we can now directly compare their components computing the square difference between the two representations:

Copy Rights

$$Dist(L_Q^{(r)}, L_I^{(r)}) = \|L_Q^{(r)} - L_I^{(r)}\|_2^2 = \sum_{i=1}^{32} (x_i^{(r)} - \overline{x}_i^{(r)})^2 + (y_i^{(r)} - \overline{y}_i^{(r)})^2. \tag{3.5}$$

The elements I_1, ... I_N of the database are ranked using *Dist()* as defined above. Figures 4 (b) and (c) show the first two images ranked by the system and the associated distance values (respectively, 0 and 6) with respect to the query shown in Figure 4 (a).

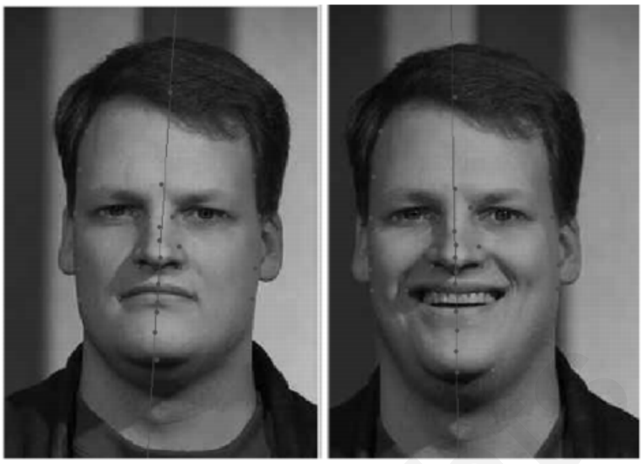


Figure 3. Two different images of the same individual with a different slope of the head (the original images have been taken from [Phillips]).

4. EXPERIMENTAL RESULTS

We have tested the system using frontal images taken from the grayscale FERET dataset [Phillips]. We have composed a test "gallery" database by randomly selecting 175 images from the FERET archive. Each image of the gallery database corresponds to a different individual. Images in the database show different facial expressions (e.g., "normal", "smiling", etc.), different lighting conditions and slightly different head orientations.

Other 75 different photos of the same individuals of the FERET database have been randomly selected and used as queries. The photo used as queries have not been included into the system's database. Figures 3 and 4 show some examples of the images used.

For each query image our system localizes 20 landmark points as shown in Section 2. The resulting landmark vector of each query has been used in order to retrieve the most similar images of the test database.

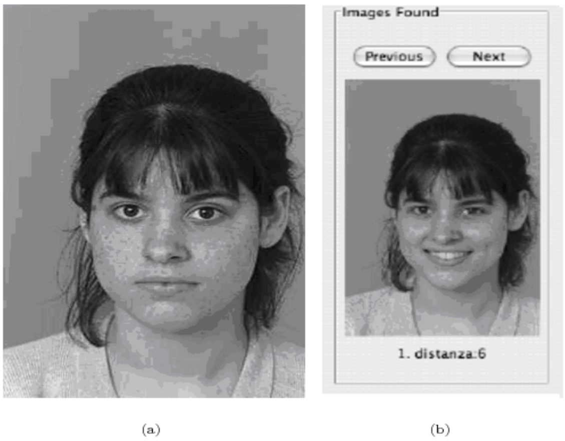


Figure 4. An example of output of the system's retrieval facilities with images taken from [Phillips]. (a) The image used as query. (b) the first image ranked by the system and its associated distance value with respect to (a).

Copy Right

Table 1 shows the results. The second row shows the number of queries for which the corresponding correct individual of our test database has been ranked in the i-th position. The third row of Table 1 shows the *Cumulative Matching Characteristic* (CMC), defined as the probability *CMC(r)* that, for a given query, the correct corresponding individual of the database is classified by the system in the first *r* positions.

As it is evident from Table 1, in more than 50 % of the queries the correct individual has been retrieved in the first 5 positions and in all the 75 performed trials the user needed to scan at most the first 13 photos shown by the system before retrieving the correct searched for individual.

Table 1. Number of queries for which the corresponding correct individual of the test database has been ranked by the system in the first *r* positions

Position	1	2	3	4	5	6	7	8	9	10	11	12	13
Number of queries	25	2	4	4	3	8	12	1	1	2	4	5	4
CMC	0.33	0.36	0.41	0.47	0.51	0.61	0.77	0.79	0.80	0.83	0.88	0.95	1

5. CONCLUSION

We have presented a face recognition approach based on the anatomic configuration of a set of facial feature points. One of the peculiarities of our system is its independence on the image type from which such points are extracted. The face landmarks are *automatically* localized by classifying SIFT feature candidates and using a modified version of the Hausdorff distance. After that, the landmark positions are used for matching with the landmark layouts extracted from the corresponding images of the gallery.

The proposed system is able to compare drawings with real images as well as images of the same person taken in different ages. This is possible because the anatomic displacements of the landmark points is distinctive for a human being and it is independent of lighting condition changes or other factors which influence the common holistic-based face retrieval approaches.

Copy Rights

REFERENCES

Bartlett, M. S., Littlewort, G., Fasel, I., and Movellan, J. R. Real time face detec- tion and facial expression recognition: Development and applications to human computer interaction. In CVPR Workshop on Computer Vision and Pattern Recognition for Human-Computer Interaction (2003).

Belhumeur, P. N., Hespanha, J. P., and Kriegman, D. J. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. IEEE Trans. Pattern Anal. Mach. Intell. 19, 7 (1997), 711–720.

Bruce, V. Recognising Faces. Chapter Faces as Patterns. Lawrence Erlbaum Associates, 1988. [Brunelli Brunelli, R., and Mich, O. SpotIt!: An interactive identikit system. GMIP 58, 5 (September 1996), 399–404.

Brunelli, R., and Poggio, T. Face recognition: Features versus templates. IEEE Trans- action on Pattern Analysis and Machine Intelligence 15, 10 (1993), 1042–1052.

D. Cristinacce, T. Cootes, and I. Scott. A multi-stage approach to facial feature detection. In British Machine Vision Conference (BMVC04), pages 277–286, 2004.

Del Bimbo, A. Visual Information Retrieval. Morgan Kaufmann Publishers, Inc. San Fran- cisco, California, 1999.

D. G. Lowe. Distinctive image features from scale-invariant keypoints. In International Journal of Computer Vision, 2004.

Gaeta, M., Iovane, G., and Sangineto, E. A survey on nonrigid object recognition approaches and their applications to face detection and human body detection. J. of Information and Optimization Sciences (JIOS), ISSN: 0252-2667 (2005).

Gillenson, M. L., and Chandrasekaran, B. A heuristic strategy for developing human facial images on a CRT. Pattern Recognition 7 (1975), 187–196.

Laughery, K. R., and Fowler, R. H. Sketch artist and identi-kit. Procedure for recalling faces. Journal of Applied Psychology 65(3) (1980), 307–316.

O. Jesorsky and K. J. Kirchberg and R. W. Frischholz. Robust face detection using the Hausdorff distance. In 3rd International Conference on Audio- and Video-Based Biometric Person Authentication, 2001.

P. A. Viola and M. J. Jones. Fast and robust classification using asymmetric adaboost and a detector cascade. In NIPS, pages 1311–1318, Vancouver, British Columbia, Canada, 3–8 December 2001.

Pentland, A., Picard, R. W., and Sclaroff, S. Photobook: Tools for content-based manipulation of image databases. In SPIE, Storage and Retrieval Image and Video Databases II (1994).

Phillips, P., Wechsler, H., Huang, J., and Rauss, P. The FERET database and evaluation procedure for face recognition algorithms. Image and Vision Computing 16, 5 (1998), 295—306.

Rowley, H. A., Baluja, S., and Kanade, T. Neural network-based face detection. IEEE Trans. Pattern Anal. Mach. Intell. 20, 1 (1998), 23–38.

Shepherd, J., and Ellis, H. Face recognition and recall using computer–interactive methods with eye witnesses. In Vicki Bruce and Mike Burton, editors, Processing Images of Faces, chapter 6. Ablex Publishing Corporation Norwood, New Jersey (1992), pp. 129–148.

Turk, M., and Pentland, A. Eigenfaces for recognition. Journal of Cognitive Neuroscience 3, No. 1 (1991), 71–86.

Vivek, E. P., and Sudha, N. Robust Hausdorff distance measure for face recognition. Pattern Recognition 40, 2 (February 2007), 431–442.

Yang, M.-H., Ahuja, N., and Kriegman, D. J. Face detection using mixtures of linear subspaces. In FG (2000), pp. 70–76.

Yang, M.-H., Kriegman, D. J., and Ahuja, N. Detecting faces in images: A survey. IEEE Trans. Pattern Anal. Mach. Intell. 24, 1 (2002), 34–58.

Copy Rights

Copy Rights

Short Papers

Copy Rights

Copy Rights

DESIGN OF A MULTIPLE-SERVER SYSTEM FOR COOPERATIVE LEARNING AND EMERGENCY COMMUNICATION

Yoshio Moritoh *, Yoshiro Imai**, Hitoshi Inomo**, Shigeaki Ogose**,
Tetsuo Hattori** and Wataru Shiraki**
**Kagawa Junior College , 10 Hama-ichiban-cho Utazu-cho Ayauta-gun, Kagawa, 769-0201 Japan*
***Faculty of Engineering, Kagawa University , 2217-20 Hayashi-cho Takamatsu city Kagawa Pref. 761-0396 Japan*

ABSTRACT

A distributed multiple server system is designed and implemented with Web-DB based services, which can play an important role not only to provide an environment for cooperative learning but also to support a function for emergency communication. In many instances, such an environment or a function used to be designed as so-called dedicated system, which can perform only single purpose. In other words, these different functions frequently seem to be mutually exclusive so that they may be realized independently with absolutely different methodologies. In our case, however, two different specifications have been accomplished by one identical system. The system has employed multiple servers located in a distributed campus network environment. Each server has multi-core processors. With virtualized CPUs by server virtualization, some programs are executed in parallel (on the virtual servers) so that our system can efficiently perform several functions. Based on our related works, two major applications are realized on the system. It can provide a cooperative learning environment for educational tool as well as Web-based surveillance functions for emergency contact.

KEYWORDS

Distributed multiple server system, Web-DB based service, Cooperative learning, Emergency communication.

Copy Right

1. INTRODUCTION

Nowadays, it becomes very much necessary for several types of users to take advantages of efficient information exchange among many distributed systems, such as network servers, control system, educational system and so on. And it is also important to design and achieve more suitable mechanism for cost-effective services of information sharing and exchanging environment. There are many researches to propose and provide educational systems in order to utilize distributed cooperative learning environments [1][2].

In the case of ourselves, for example, we have already obtained good analytical results for our educational tool in the cooperative learning field through real education. Based on the above successful background, we have been going to design and implement information server system in order to realize a distributed information-processing environment for cooperative learning. This system employs a configuration of distributed environment with multiple servers connected and located in the three campuses initially. In addition, by means of employment of some suitable schema, it is possible to provide both effective structures of cooperative learning and efficient methods of emergency contact with information exchange concurrently. In real education, such a strategy may be very much useful to maintain practically robust schooling.

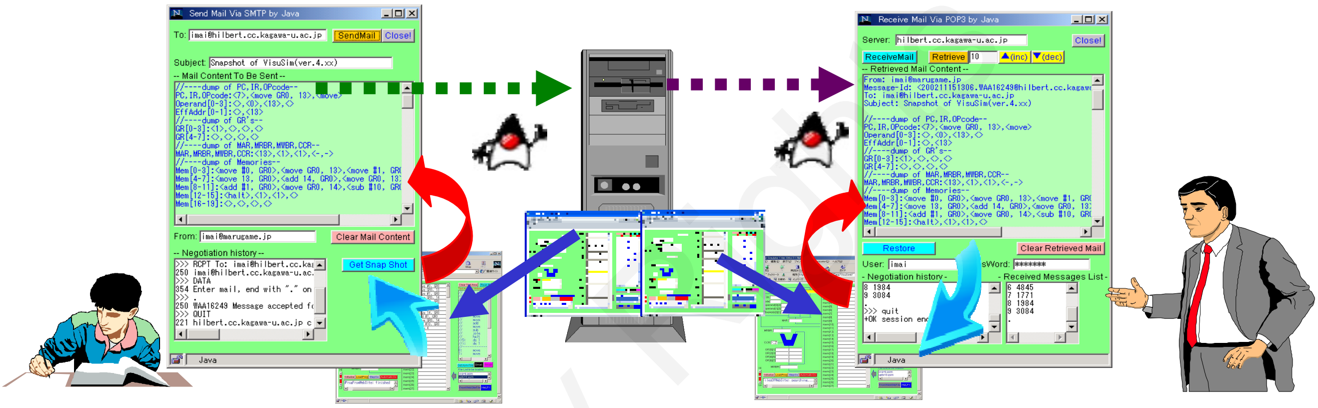
This paper describes a distributed multiple server system for cooperative learning at normal times as well as emergency contact with out-of-hours communication. It explains our related works for help to develop our new system in the next section. It introduces design concept of our special-purpose server system and illustrates its system configuration and development in the third one. It describes some applications with such a system in the fourth one. And finally it summarizes some conclusions and future problems in the last one.

2. OUR TYPICALLY RELATED WORKS

Case(I): Visual Computer Simulator.

In order to learn computer system, it is very important to understand the internal behavior and structure of computer. In such a case, it is effective for a learner to utilize an educational tool with visualization facility. For example, a learner uses the educational tool to simulate a sample program graphically in the register-transfer level. If a learner wants to know a role of a specific register, PC (Program Counter), he or she can stop the simulation at any point, change the value of PC, and then restart the simulation in the register-transfer level. An effect of simulation increases more and more with such a visualization facility.

When learners use the educational tool and want to correspond with others, they can invoke e-mail sending or receiving module of the tool. These client modules are available whenever the tool is invoked as a Java stand-alone application as well as a Java applet. An SMTP-client is a Simple Mail Transfer Protocol-based sending module, which transfers message from the tool to the server. Figure1 shows a schematic relation between SMTP-client, POP3-client and our educational tool. This figure illustrates a typical story that a learner downloads the tool, invokes SMTP-client, adds the current state of the tool into an e-mail, and easily send such an e-mail to the mail/Web server.


Send Mail Via SMTP by Java
SendMail
Close!
-- Mail Content To Be Sent --
Clear Mail Content
-- Negotiation history --
Get Snap Shot
Java
Receive Mail Via POP3 by Java
Close!
ReceiveMail
Retrieve
▲(inc)
▼(dec)
-- Retrieved Mail Content --
Restore
Clear Retrieved Mail
- Negotiation history -
- Received Messages List -
Java

To: imai@hilbert.cc.kagawa-u.ac.jp

Subject: Snapshot of VisuSim(ver.4.xx)

```
//----dump of PC,IR,OPcode--
PC,IR,OPcode:<7>,<move GR0, 13>,<move>
Operand[0-3]:<>,<0>,<13>,<>
EffAddr[0-1]:<>,<13>
//----dump of GR's--
GR[0-3]:<1>,<>,<>,<>
GR[4-7]:<>,<>,<>,<>
//----dump of MAR,MRBR,MWBR,CCR--
MAR,MRBR,MWBR,CCR:<13>,<1>,<1>,<-,->
//----dump of Memories--
Mem[0-3]:<move #0, GR0>,<move GR0, 13>,<move #1, GR
Mem[4-7]:<move 13, GR0>,<add 14, GR0>,<move GR0, 13
Mem[8-11]:<add #1, GR0>,<move GR0, 14>,<sub #10, GR
Mem[12-15]:<halt>,<1>,<1>,<>
Mem[16-19]:<>,<>,<>,<>
```

From: imai@marugame.jp

```
>>> RCPT To: imai@hilbert.cc.kag
250 imai@hilbert.cc.kagawa-u.ac.
>>> DATA
354 Enter mail, end with "." on
>>> .
250 WAA16249 Message accepted fo
>>> QUIT
221 hilbert.cc.kagawa-u.ac.jp c
```

Server: hilbert.cc.kagawa-u.ac.jp

Retrieve: 10

```
From: imai@marugame.jp
Message-Id: <200211151306.WAA16249@hilbert.cc.kagawa
To: imai@hilbert.cc.kagawa-u.ac.jp
Subject: Snapshot of VisuSim(ver.4.xx)

//----dump of PC,IR,OPcode--
PC,IR,OPcode:<7>,<move GR0, 13>,<move>
Operand[0-3]:<>,<0>,<13>,<>
EffAddr[0-1]:<>,<13>
//----dump of GR's--
GR[0-3]:<1>,<>,<>,<>
GR[4-7]:<>,<>,<>,<>
//----dump of MAR,MRBR,MWBR,CCR--
MAR,MRBR,MWBR,CCR:<13>,<1>,<1>,<-,->
//----dump of Memories--
Mem[0-3]:<move #0, GR0>,<move GR0, 13>,<move #1, GR
Mem[4-7]:<move 13, GR0>,<add 14, GR0>,<move GR0, 13
Mem[8-11]:<add #1, GR0>,<move GR0, 14>,<sub #10, GR
Mem[12-15]:<halt>,<1>,<1>,<>
```

User: imai
sWord: ********

```
8 1984
9 3084
.
>>> quit
+OK session en
```

```
6 4845
7 1771
8 1984
9 3084
.
```

Figure 1. Information sharing with communicating functions of visual computer simulator

Our educational tool provides simple information-transmission service which can send a message including the current state of the tool, namely, all the contents of registers and memory of itself, to a mail server. With the above service using SMTP-client, the tool can make the copy of itself onto another instance of it. There is a particular condition or limitation about capability of SMTP-client. A message from the tool must be transferred to the mail server. Namely, it has to access the external server which is out side of the machine where it executes. When it works as a Java applet, its SMTP-client should be invoked as a child process of such an applet so that it can only communicate with the mail server which is the same machine of Web server of the tool itself.

On the other hand, a POP3-client is a Post Office Protocol-based receiving module, which obtains information message from the server to the tool. Of course, as described above, there is the same condition for POP3-client to access the server just like STMP-client does. Two users of our educational tool can communicate with each other, throw a message from one to another through the intermediary of the mail server, and finally share the same information related to educational tool between each other. This figure also illustrates another typical story that an instructor asynchronously downloads the tool, invokes POP3-client, receives an e-mail with other internal status of the tool, restore such a status onto own his tool, and readily shares the same status of other tool of learners.

It will be a good example of useful interdependence. In such a case, each learner is held accountable for doing his or her share of given problems and for mastery of all of the jobs to be learned. In order to exchange frequently several information and idea between each other, facility of Visual Computer Simulator, such as function of communication support, is very efficient and effective for a pair of learners to study and exercise assembly programming by themselves in the above cooperative learning environment.

Case(II): Web-based surveillance System

An information server has been developed to work as the kernel of Web-based surveillance system. Web, mail, and database facilities are integrated in the server function. The picture information is obtained from the network camera, accumulated in the server, and distributed to its clients according to their requests. In addition to the remote monitoring function, our server can provide a service for household appliance control.

As a GUI client, the server has utilized some kinds of cellular phones, and CLDC-based Java programming has been employed to realize the function of clients. For the sake of enhancement of our monitoring function, change from an acquisition picture to another can be analyzed by means of image processing. A service of urgent connection between clients and server is also adopted based on the result of image processing. Figure2 shows our surveillance system and high-performance cellular phone as its client.

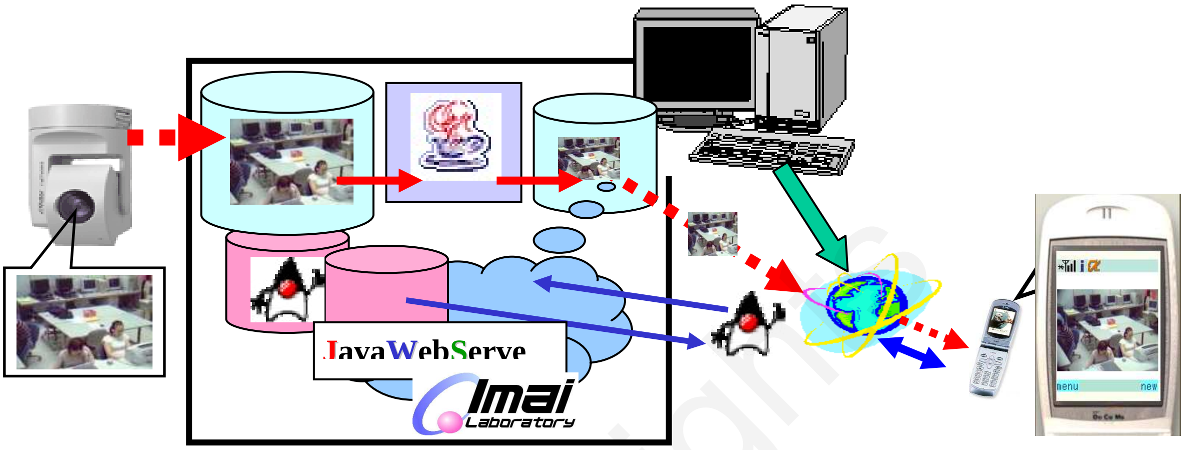


Figure 2. Schema of GUI utilizing high-performance cellular phone of our web-based surveillance system

3. SYSTEM CONFIGURATION AND DEVELOPMENT

Our university has four major campus and six faculties and several institutes/centers. They are connected one another by means of dedicated high-speed network. Tele-conference and remote education can be performed with interconnection by such a network. In order to keep network security in the university level, we have the (external, first-level) firewall at the connected point with the Internet backbone and some UTMs (Unified Threat Managements; namely, internal (second-level) firewalls) at the interconnected points for the all the campuses. And we have anti-virus mail servers and spam firewalls (i.e. Barracudas) in addition. In the case to carry out remote education, we have introduced some educational systems in our university. It is necessary to operate and modify the according UTMs for smooth communication between the target campuses. In the case of tele-conference, the famous PolyComs and Live Meetings have been introduced and utilized by more and more users of our university. We have some problems in tele-conference, and they are sometimes related to file sharing, so we think that it must be important how to realize and manage remote file sharing suitably.

From the achievement record, almost users of our university depend on using WWW and e-mail services in order to perform information transfer and exchange. Not only students but also university staffs do like to utilize simple functions and manipulations in daily operations. There are additional merits because almost Firewalls and UTMs allow to pass the above HTTP/SMTP-based packets and communication without problems. So such a case is suitable for several types of network security policies, so that it can avoid needless confusion before some incidents happen. Based on the above discussion, we think that information transfer and exchange could be limited to HTTP and SMTP protocols from the viewpoints of network managements and security measures. The above HTTP includes HTTPS protocol and the above SMTP includes POP3 and IMAP4 ones, just in case. So it must be better for us to implement information transfer and exchange service based on the above protocols. Especially, employment of SMTP protocol is suitable to link up with spam firewall and other security devices.

It is necessary to have an affinity to existing network facilities when we develop new information server system and try to start its services in our real situation. Our system has employed multi-server configuration

Copy Rights

located in the distributed campuses, which will be explained in the later description. And then we must decide to choose suitable protocols and communication procedures in order to be consistent with our network facilities and security services.

From the above discussion and our experiences, our multi-server system has employed the following configurations and specifications.

a) Our system has the three or more homogenous information servers which are located in our distributed network environment.

b) Each information server can provide mirroring function against the specific server for the sake of reliable information exchange.

c) Sampling rate of mirroring function between pair of servers can be adjusted to cumulate users information and learning contents in desirable reliability level.

d) Registered users can access any server of the multiple server system so that they can choose the most convenient access point of the system.

e) Collaborative communication and information exchanging are available between users who login the system from any server.

f) Our system can inquire after its users based on users information, even if damages partly happen (for example, a server becomes dysfunctional, or it ceases to function properly in the cooperation between specific servers).

g) Our system can provide communication method such as information transfer and exchange between users at a normal condition as well as extraordinary one.

h) Our system can support user collaboration and both environment for not only cooperative learning but also emergency contact.

PC, PDAs (smart phones) and cellular phones are expected as clients of our distributed multiple server system. Each information server of the system must treat with multimedia information. It must be also designed to support cooperative learning in distributed environment, asynchronous information sharing, and emergency communication between clients at any extraordinary condition. As our previous works illustrate, we have obtained some kinds of experiences and skills from information server configuration, their applications, and continuous managements. It seems to be an effective strategy to combine and integrate individual functions into a new solution for complicated problems. So it is employed that multiple servers are located in the distributed campus network and they work together with existing facilities and services.

Figure3 shows an overview of the system configuration for our distributed multiple server system. The system has been implemented in our distributed campus network environment. And each sub-system includes rack-mounted server, UPS and several kinds of clients, such as usual PC, PDA/cellular phone and multimedia input/output devices. Sub-systems are linked and interconnected with campus network by means of Giga high speed.

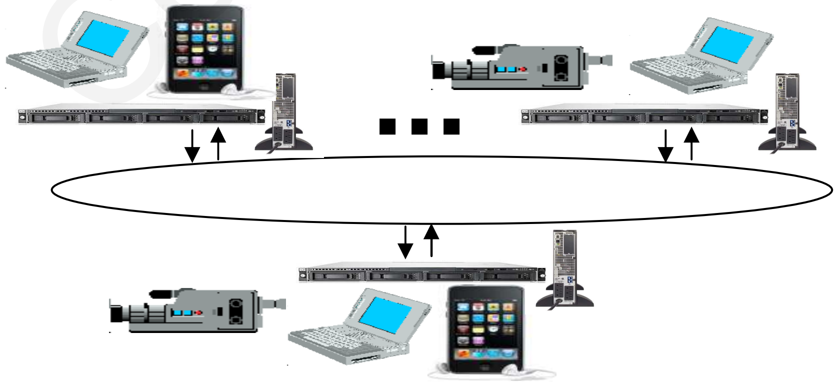


Figure 3. Configuration overview for our distributed multiple server system

Each information server also employs the architecture of multiple processor cores and virtualized CPU technology, namely Virtual Machine Scheme, named Citrix Xen Server Virtualization. It is very efficient to realize parallel programs execution and smart management of concurrent services. If some applications need

Copyrights

more powerful CPU services, it can be adjusted to migrate (assign) virtualized CPU and related resources to those specific applications according to dynamic demand changing and/or configure modification.

One of multiple processor cores has been designed to execute most useful Linux-based Web-DB software fundamentals. This is a basic layer (i.e. platform) for usual and classical applications. One of others is sometimes assigned to multimedia information processing modules, such as image understanding, video transmission, voice generation and so on. Some of them are potentially adjusted to carry out asynchronous information sharing functions and support for emergency contact and/or urgent situation changing.

This approach coincides with our basic principle for realization of information transmission and exchanging with HTTP and/or SMTP protocols. In general, the real educational situation needs Windows server and Windows-based information sharing so that our system must prepare for the according requests and specification. The system may provide some platforms in order to let MS-Windows server software be executed in one of multiple processor cores.

With such a mechanism described before, it is possible for system designer/developer to configure one information server including two or more functions and/or services, which clearly differ from one another in their objectives to realize. Each function or service can be executed sometimes in parallel and concurrently. In the former case, function or service is assigned and executed in the different processor, while they are assigned to the single processor and executed in turn in the latter case.

4. SOME APPLICATIONS OF MULTI-SERVER SYSTEM

Case(I): Cooperative Learning by means of Our System

In order to manage our educational tool effectively, it is indispensable to design and implement a special-purpose information server which can provide some kinds of information-exchanging environment for the tool and its users. With such a server, the tool can play the very important role to carry out communication among users. Built-in e-mail handlers of the tool realize such communication between users .

Learners using our tool, visual computer simulator, can obtain their necessary information from the special-purpose information server through its communication supporting functions according to their understanding levels. So the information server needs the following three basic functions.

- Web service function: They are very much essential to deliver the program (executable) code of Simulator and sample (source) programs for Simulator. They correspond to HTTP-based communication with 3-way hand-shaking procedure. Additionally, they support FTP-based data transferring service.
- e-Mail service function: Simulator can support the communication and information-sharing mechanism among users by means of SMTP-based and/or POP3-based facilities. It is necessary for the server to be implemented to provide SMTP-transferring function and POP3-receiving ones.
- User management service function: There must be user management functions in the server, not only because of POP3-service but also because of user identification to recognize user's understanding level. The former is necessary to realize POP3-service, while the latter is essential to specify user's level to utilize Simulator more effectively.

With these functions, the special-purpose information server prepares necessary and minimal conditions to realize communication supporting and information-exchanging environment for the educational tool.

a) Cooperation among multiple servers: The previous information server simply employed one server system for all the users' management, user identification with serial number, so that there were some regulations such as not so good two-way authentication among users, not so efficient information transmission and/or sharing between different level of users, etc.

Now we have employed a new resolution to group all the users into cluster and assign such a cluster to one server as one of temporary expedients for management of users among multiple servers. Of course, the above palliative treatment is not effective for cooperation among multiple servers. It is necessary to establish more effective methods to organize different servers. One of those is to utilize e-mail function of the educational tool and communication facilities between its users. An then all the users of the tool are registered on the shared user database and their mail spool area and users' home directories are created in the shared volume area by means of NIS/NFS or SAMBA (or NAS) facilities.

b) User management for multi-servers: Flexible user identification is necessary to allow hierarchical user naming. With introduction of LDAP (Lightweight Directory Access Protocol) based authentication, it is very

Copy Rights

much smooth to manage the users among multiple servers and easy to implement flexible user authority for such servers. Although this is a useful method for user management, it will be suffering from some dangerous intrusion without closed network based characteristics and benefits. Security problems are very much heavily serious and expensive to protect correctly and emergently. Additional facilities such as NAT/NAPT (Network Address (& Port) Translation) mechanism will be implemented into a new information server simultaneously.

Case(II): Surveillance and Emergency Contact through System

One server obtains image from network camera and cumulates it into Database. The image is sometimes one of object for image processing in the server. For example, one server performs image processing for a series of sampling images by means of background difference method. In order to judge whether there is an intruder or not at the target place, the server can compute area of intruder which is recognized as difference of the pre-defined basic frame, compare the target image with such a basic frame and finally decide whether an intruder comes or not.

When image processing procedure points out the difference between compared images, the according server has sent e-mail to other servers as well as the previously entered clients. Especially, it is effective to send e-mail to cellular phones because almost all users always carry such phones with themselves. Some emergency contact can be performed through such message mailing service.

5. CONCLUSION

In the case of application for cooperative learning, the system can realize multiple users collaboration through configuration of multiple servers in a distributed environment. It facilitate asynchronous information exchange among the multiple servers, because it employs virtual sever mechanism and always invokes communication-oriented procedure. In other case of application, our system can assign and run limited surveillance program into one of its virtual servers. Such a program always monitors its target place, recognize it whenever registered conditions happen at the place, and transfer such information to the specified clients by means of mobile communication. The above applications can be easily installed and work suitably in the multiple server system. So we can summarize our work as follows;

1) Our multiple server system employs a virtual server scheme with Xen mechanism so that several applications can work easily and concurrently.

2) An educational tool can be executed in one of such virtual servers together with Web and DB services. And it provides cooperative learning environment effectively.

3) At the emergent condition, multiple server system can automatically switch surveillance program into foreground and realize emergency contact and transfer suitable information between specified users.

Copy Rights

ACKNOWLEDGEMENT

This study was partly supported by Grand-in-Aid for scientific research from the Ministry of Education, Culture, Sports, Science and Technology under the contract No. 19500044. And it was also partly supported by Research for Promoting Technological Seeds from Japan Science and Technology Agency (JST).

REFERENCES

[1] Shen, J., Hiltz, S.R. and Bieber, M., 2006. Collaborative Online Examinations: Impacts on Interaction, Learning, and Student Satisfaction. *In IEEE Transaction on Systems, Man, and Cybernetics—Part A: Systems and Humans,* Vol.36, No.6, pp.1045-1053.

[2] Nguyen, D., Guggisberg, M. and Burkhart, H., 2006. CoMobile: Collaborative Learning with Mobile Devices. *Proceedings of the 6th IEEE International Conference on Advanced Learning Technologies,* Kerkrade, Netherland, 5 pages.

IMPACT OF GATEWAY DISCOVERY ON TCP-CONNECTIONS IN MANETS

A. Triviño Cabrera, M. C. González Linares, E. Casilari and F. J. González Cañete
*Dpto. Tecnología Electrónica, Universidad de Málaga **
*Málaga, Spain**

ABSTRACT

Mobile ad hoc networks (MANET) can be formed in scenarios where the access to the Internet is demanded. Most of the Internet traffic could be supported by TCP traffic (e.g. HTTP traffic). However, TCP technology needs to be adapted to cope with the different features of the wireless (the MANET node) and the wired (Internet host) mediums. The functionalities for this adaptation could be implemented in the Gateway as it is proposed by TCP-GAP. This new version of TCP is specially designed for MANETs connected to the Internet. Although the evaluation results seem promising, this new technology has been exclusively evaluated in static scenarios. This kind of scenarios represents a simplification of real MANETs where terminals can freely move. Furthermore, static scenarios do not deal with the problem of acquiring or updating the route to the Gateway which connects to the Internet. Our paper focuses on the evaluation of TCP-GAP in mobile environments. In particular, it analyzes the impact of the gateway discovery procedure on the TCP performance by means of simulations. As the simulations show, the gateway discovery procedures, by which mobile nodes update the route to the Gateway, have a significant effect on the asymmetry of the TCP-flows.

KEYWORDS

MANET, Internet, Multihop, wireless, Gateway, Mobile.

1. INTRODUCTION

Mobile Ad hoc NETworks (MANET) are composed of wireless devices that communicate without any infrastructure. The communication between two distant nodes (not directly connected) is achieved by intermediate nodes which collaborate in the retransmission of the packets from the source to the final destination. Due to the capability of offering connectivity without any deployed equipment, this technology was initially conceived to work with in military scenarios or environmental disasters. However, the popularity of mobile devices and the growing interest of being connected anywhere and anytime have extended this technology to other applications. For instance, a MANET can be used in a visiting park or a conference venue in an economic way. If we pay attention to these new applications, we can expect that users demand access to the Internet. Furthermore, most of the access is based on TCP (Transmission Control Protocol) connections. In order to interconnect a MANET to the Internet, a conventional Access Router is not sufficient. The Access Router emits Router Advertisement messages [Narten, T et al, 2007] which cannot be propagated as they are generated with link-local addresses [Hinden, R.et al, 2006]. However, these messages are needed by all the nodes in a MANET as they contain the information necessary to enable mobile nodes to generate or acquire an IP address. As a MANET node could be several hops away from the Access Router, a new element is required in order to guarantee that all the nodes know the configuration parameters. This new element is a Gateway. The Gateway is directly connected to the Access Router via a wireless or wired link. It is responsible of generating Modified Router Advertisements (MRA), which are a modified copy of the Router Advertisements created by the Access Router. The main difference is that MRA can be forwarded so all the nodes in the MANET can receive them. Besides, the reception of the MRA messages is used by mobiles nodes to create/update or optimize the route to the Gateway, that is, the route to send the packets to the Internet. Taking into account this use, the question is when MRA messages should be generated.

The process by which MRA messages are created differentiates the gateway discoveries. In the global connectivity support [Wakikawa, R. et al, 2006], which is the most popular mechanism to integrate a

MANET into the Internet, there are three types of gateway discovery: reactive, hybrid and proactive. Although some studies have already focused on the impact of the gateway discovery on UDP transmissions, this paper analyzes the TCP performance for the three gateway discovery procedures. In particular, the studied TCP technology includes some techniques to adapt its congestion control to the different mediums (wireless, wired) in an Internet-connected MANET. It is called TCP-GAP (Gateway Adaptive Pacing) [ElRakabawy, S. M. et al, 2008]. TCP-GAP has already been studied in [ElRakabawy, S. M. et al, 2008]. However, these studies are restricted to static scenarios. In this kind of environments, the routes to the Gateway are always active and they are assumed to be statically configured. Thus, the protocol has been evaluated without taking into account the effects of the gateway discovery procedure. Our paper provides an analysis about the performance of TCP-GAP for different gateway discoveries in mobile scenarios. As shown, the main effect of the type of Gateway discovery is the asymmetry provoked on the TCP-flows.

The rest of the paper is structured as follows. Section 2 describes the TCP-GAP protocol. Section 3 explains the functionality of the Global Connectivity support. Section 4 shows the simulation results and the explanations about this performance. Finally, Section 5 draws the main conclusions of our work.

2. TCP IN INTERNET-CONNECTED MANETS

TCP is a technology that offers a reliable transmission for upper layers. In order to guarantee this reliability, each data segment is tagged with a sequence number. The receptor confirms the reception of the datagram using the sequence number in a specific message called ACK (Acknowledgment message). Cumulative acknowledgment is possible so multiple segments can be confirmed with just one ACK message. Additionally, flow and congestion control are also present in TCP technology. Flow control is supported by a window that represents the interval of segments that can be sent while the sender is waiting for their confirmations. Alternatively, the congestion control stands up for the algorithms that avoid congestion in the network provoked by the repeated emission of non-confirmed segments [Allman, M. et al, 1999].

TCP technology is based on the assumption that losses are due to congestion problems. This condition holds in wired networks but in wireless connections losses are mainly caused by interferences. In order to adapt TCP to the wireless scenarios, several versions of TCP have been proposed [Tian Y. et al, 2005]. In the context of Mobile Ad Hoc Networks, TCP must also cope with multihop communications. [Sundaresan, K. et al, 2003] [ElRakabawy, S. M. et al. 2005] are some protocols specifically proposed for MANETs. On the other hand, TCP-GAP focuses on the fact that an Internet-connected MANET deals with the two different mediums [ElRakabawy, S. M. et al, 2008]. To improve the goodput offered by the network, an adaptive transmission rate is used in the mobile nodes and in the Gateway. Assuming 802.11-based connections, the protocol states that the interferences in a single hop can be avoided if a data segment is transmitted when the previous one has already reached the next four hops (this condition is set taking into account the hidden terminal effects). Therefore, the estimation of the four-hop propagation delay (FHD) of TCP segments constitutes the basis of the algorithm that adapts the transmission rates.

In a similar way, the Gateway adapts its transmission rate to the estimated FHD. In contrast, this estimation is particular for each flow. The Gateway stores the received segments in an internal buffer and retransmits them according to the estimated rate (which is based on the measured RTT for each flow). For more details about the setting of the transmission rate, please refer to [ElRakabawy, S. M. et al, 2008].

3. GATEWAY DISCOVERY

The key element to connect a MANET to the Internet is a Gateway. The gateway has two main functionalities. Firstly, it distributes the configuration parameters in the network. This is done by the emission of specific messages called Modified Router Advertisement (MRA) messages. Additionally to the acquisition of the configuration parameters, these messages are employed by the mobile nodes to create, update and/or optimize the route to the Gateway, that is, to the Internet. Additionally, the Gateway routes the packets that the Access Router receives in the MANET. Conventional Access Routers do not implement any ad hoc routing protocol so a Gateway is necessary to discover and route the packets in the MANET.

The Global Connectivity support is the most popular scheme to integrate a MANET into the Internet. In this mechanism, three gateway discoveries are described: reactive, proactive and hybrid. In the reactive scheme, the mobile nodes that need to send a packet to the Internet and it does not hold a valid route to the gateway, generates a Modified Router Solicitation (MRS) message. This message is broadcast in the network and when received by the Gateway, it replies with a unicast MRA message. This response is propagated just in the path from which the Gateway has received the MRS. In contrast, the proactive gateway discovery is based on the periodic emission of broadcast MRA messages each *T* seconds. Thus, mobile nodes periodically receive a copy of a MRA message and, in turn, periodically update the route to the Gateway. Along the interval of emission of the MRA message, the mobile can freely move. This movement could make the stored routes stale so a mobile node needing to route the packets to the Internet acts reactively. Finally, the hybrid scheme combines the two previous mechanisms. In an area close to the Gateway, MRA messages are periodically broadcast. However, those nodes that are outside this area (defined by the TTL parameter) are forced to know the routes to the Gateway in a reactive way.

4. EVALUATION OF THE IMPACT OF GATEWAY DISCOVERY

The goal of this paper is to evaluate the impact that the type of gateway discovery provokes on TCP performance. For this task, we use a comparison approach so that the goodput offered by TCP-GAP is measured for the three types of gateway discovery (reactive, proactive and hybrid) under the same conditions. In order to repeat the experiments under the same environmental settings, we have conducted our study by means of simulations. In particular, NS-2 was employed [Fall, K. et al, 2010].

The tests were performed in a squared area of 1000x1000 m^2. The gateway is placed in the center of the area, that is, at (500,500) m. One TCP connection is established in every simulation. The TCP traffic is from a mobile node to the Internet host which is acceded through the Gateway. In particular, an FTP connection is set. Additionally, 5 CBR exchanges are incorporated into the simulations. They emit 4 packets/s and the packets have 512 Bytes. This inclusion introduces interferences so TCP-GAP can be evaluated in more realistic scenarios.

Mobile nodes are equipped with 802.11 compatible radio interfaces. The transmission range is set to 250 meters. The interference radius is 550 m and the propagation model used in the simulations is the Two-Ray Propagation model. Concerning the mobility of the nodes, they follow a Random WayPoint (RWP) mobility pattern. This is a well-known mobility model used in MANETs. Every mobile node randomly chooses a destination. Once selected, it goes to this point at a constant speed. The speed is also randomly computed from a minimum value to a maximum value. A uniform distribution function is assumed for this computation. Once the destination is reached, the mobile node stays in that position for a pause time. Then, it selects a new destination and it repeats the process. As recommended in [Yoon, J. et al, 2003], the minimum speed is set to 1 m/s. The mobility parameters and other simulation variables are listed in Table 1.

Table 1. Simulation parameters

Simulation Parameter	Value
Area	**1000x1000 m^2**
Number of mobile node	**50**
Mobility Pattern	**Random WayPoint** **Maximum Speed: [1, 5, 10] m/s** **Minimum Speed: 1 m/s** **Pause time: 0 s**
Transmission range	**250 m**
Simulation Time	**5000 s**
Runs per simulation	**40**
Gateway Discovery	**T : 5 s** **TTL : 3**
Ad hoc routing protocol	**AODV** **Local repair disabled** **Link Layer Detection enabled**
MAC layer	**802.11 b** **RTS/CTS enabled**

Figure 1, 2 and 3 show the mean values obtained for the simulations for the three studied gateway discoveries. For each maximum speed, the tests were executed 40 times. The goodput represents the transmission rate perceived by the application layer. We differentiate between the uplink connection (from the mobile node to the Internet) and the downlink connection (from the Internet to the mobile node). In order to suppress the effects of the wired section, the host to which the data packets are sent is assumed to be directly connected to the Internet Gateway and an ideal connection with no losses is established between these two extremes.

In Figure 1, we have the results for the Reactive Gateway Discovery. In this case, the routes to and from the Gateway are discovered reactively so slight differences are perceived in the goodput of the uplink and the downlink connection. That means that the transmission rates at the mobile nodes and at the Gateway are set with similar values. The increment of the speed leads to more frequent route breakages so the goodput is reduced in highly dynamic scenarios.

On the other hand, Figure 2 shows the results when the Proactive Gateway discovery procedure is used. Under these circumstances, the routes from the mobile nodes to the Gateway are periodically updated so the perceived four-hop delay (FHD) is different in both senses. Due to this difference, the transmission rate is higher at the mobile nodes than at the Gateway. Thus, the goodput offered in the uplink connection is greater than in the downlink connection. Therefore, an asymmetrical performance is identified when the proactive gateway discovery is used.

Concerning the downlink connection, the proactive gateway discovery outperforms the reactive scheme. This is explained by the reduction of the solicitations that the Gateway must manage. This reduction represents a lower level of interferences and, in turn, an improvement in the TCP performance.

Finally, the hybrid gateway discovery (represented in Figure 3) turns into an intermediate performance. It outperforms the goodput offered by the reactive scheme but a symmetrical behavior is hold.

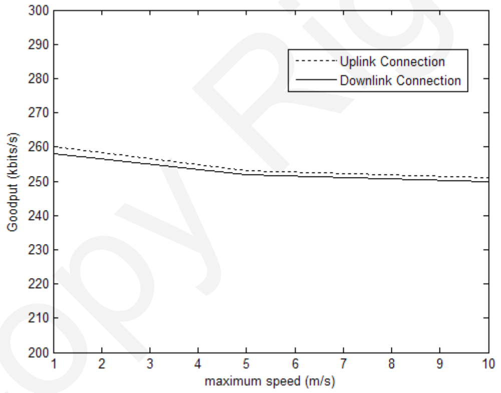


Figure 1. Goodput obtained in the reactive gateway discovery procedure.

Copy Rights

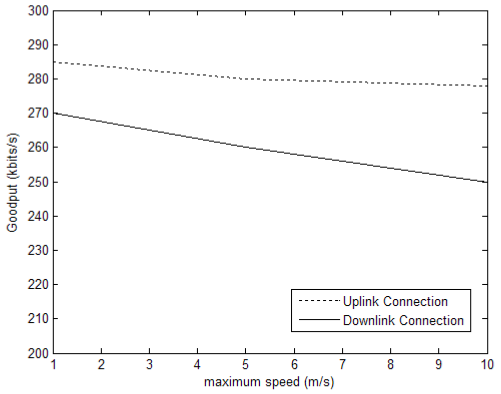


Figure 2. Goodput obtained in the proactive gateway discovery procedure.

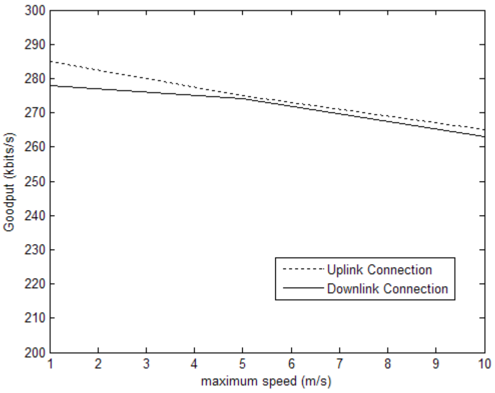


Figure 3. Goodput obtained in the hybrid gateway discovery procedure.

5. CONCLUSIONS

This paper analyzes the performance of TCP connections in Internet-connected MANETs. In particular, the TCP-GAP is used for the TCP communications. This protocol adapts its transmission rate to the congestion status of the network. The algorithm for the adjustment is different in the Gateway (which connects the MANET to the Internet) and in the mobile nodes. They are supported by the estimation of the four-hop propagation delay of the transmitted segments. Although this algorithm has already been studied in static scenarios, this delay is also affected by the time elapsed in the route discovery procedures. Therefore, a correct evaluation of the performance of TCP-GAP should comprise different gateway discovery procedures in mobile environments. This paper evaluates the impact of the gateway discovery and analyzes why this happens. We conclude that the gateway discovery procedure, which is in charge of updating the routes to the Gateway, represents an important effect on the goodput offered by TCP-GAP.

Copy rights

REFERENCES

Allman, M. et al, 1999. *TCP Congestion Control*. RFC 2581.

ElRakabawy, S. M. et al. 2005. TCP with Adaptive Pacing for Multihop Wireless Networks. *Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing MobiHoc'05*.

ElRakabawy, S. M. et al, 2008. *TCP with gateway adaptive pacing for multihop wireless networks with Internet Connectivity*. Elsevier Computer Networks vol. 52, pp. 180-198.

Fall, K. et al, 2010. *The NS manual*, VINT project.

Hinden, R.et al, 2006. *IP Version 6 Addressing Architecture*. IETF RFC 4291.

Narten, T et al, 2007. *Neighbor Discovery for IP version 6 (IPv6)*. IETF RFC 4861.

Sundaresan, K. et al, 2003. ATP: A Reliable Transport Protocol for Ad-hoc Networks. *Proceedings of the4th ACM International Symposium on Mobile Ad Hoc Networking and Computing MobiHoc'03*.

Tian Y. et al, 2005. *TCP in wireless environments: problems and solutions*. IEEE Radio Communications, March 2005.

Yoon, J. et al, 2003. Random waypint considered harmful. *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies* (INFOCOM), vol. 2, pp. 1312-1321.

Wakikawa, R. et al, 2006. Global Connectivity for IPv6 Mobile Ad Hoc Networks, IETF Internet Draft.

ENHANCED VOIP SOLUTION FOR CONTROLLING A ROBOT-COMPANION

Z. Mapundu* and Th. Simonnet**
**F'SATIE at Tshwane University of Technology, PB x680 0001 Pretoria – South Africa*
***ESIEE-Paris, BP 99 93162 Noisy-le-Grand cedex – France*

ABSTRACT

In recent years, we are witnessing a gradual migration of traditional telephone service network i.e. Public Switched Telephone Network (PSTN) world to the Internet Voice over Internet Protocol (VoIP). This phenomenon is playing a major role in telecommunication developments due to its advantages, operational infrastructure simplification and significant reduction in the cost of communication services that are linked to them. The likelihood of implementing a videoconferencing tool will be helpful for elderly people to have connection with their families, Robot-Companion and medical professionals especially in the field of trust relationship concerning their health condition. This paper describes how VoIP solution can be used to advance communication process by means of controlling a robot machine and this will involve the possibilities of integrating and testing the existing Open-Source tools.

KEYWORDS

Robot-Companion, Videoconferencing, Remote-Control, and Telemedicine.

Copy Rights

1. INTRODUCTION

European Scientists have found three new major genetic links to pre-Alzheimer, affecting up to 20% of people with brain-wasting disease and it was the most significant such discovery in 15 years. Alzheimer disease affects more than 26 million people globally and it has no cure with any good treatment and the need for effective remedies is pressing on, with the number of cases estimated to go beyond 100 million by 2050 (Commission of the European Communities, 2009**)**. Once more, states that various European projects focused on assistive technologies for helping elderly people are on progress, QuoVadis and CompanionAble are part of these projects. All these projects have a common objective, to support the elderly in daily life by integrating the existing technologies for managing their domestic ambient environment in order to increase their autonomy, safety and improve their quality life. Presently the European government is also dedicated in providing the health-care support for elderly patients to all European citizens, not as an advantage but as a fundamental right for many citizens who lacks the most basic service.

To accomplish this goal, the government and private health-care sectors have identified Videoconferencing as a strategic tool to improve the heath-care delivery and instructive services to elderly patients with the aim of reducing medical care costs. This tool is a telecommunication service for both audio and video that can be connected together in real time with many participants located in different places. Meeting by videoconferencing may vary from simple conversation between two or more people in two locations (point-point) to complex conference that connect multiple users in multiple sites (multi-point). With the advancement of Information and Communication Technology, the Telemedicine has emerged as technology based services for different health-care services (from primary to specialty or super-specialty), digital investigations, complex interpretation (radiology, pathology), epidemiology control, medical communication and etcetera in order to provide a health-care support to patients. This paper is proposing a Remote control prototype that can by used to control the robot machine and this will be achieved by using the existing OpenSource tool for testing and modification possibilities.

1.1 Related Work

Subsequently in response to the above problems, ESIEE-Paris operates and maintains videoconferencing tools that offers communication and assistance services to patient, especially for elderly people. This is a VoIP solution that has two main components, namely a Central Server and Local Equipment for Domestic Internet Gateway (DIG). Both these components uses a secured IP Network over Internet (VPNs) and according to ESIEE Paris team (Couet et al. 2008), this solution is easy to deploy because all functions and related virtualized servers can be held on one physical server.

The aim is to integrate and advance the communication process for elderly patients in order to keep a strong social link with their family members or caregivers and medical professionals, thus a high-quality videoconferencing services is implemented as a solution. ESIEE-Paris (Simonnet, 2009) has identified that the main challenge from previous research project is to manage communication between a distant operator and the robot-companion, thus an operator will need to have the ability of taking control of the robot. The control of robot-companion can be carried out in a secured wired, wireless network environment, thus will allow a realistic time control that can be achieved through the declarative definition of network components and object-oriented approach to model development and data management. Internet communication can produce an efficient and effective control solution for the robot and this approach will need to integrate the robot machine as a virtual and automated user which can be called from a classical softphone like Ekiga. Then an operator will have to call the robot and control it by sending tones online through the use keyboard or specific peripheral like joystick in order to mange the communication process between a distant operator and the robot.

A. Bley (Bley, 2008) is a Project Managing Director of MetraLabs in Germany, defines few advantages of developing such Robot-Companion:

- Real interaction partner: an embodied, anthropomorphic system with natural interface and human-like behavior.
- Embodiment guarantees visible intimacy and privacy like by closing the eyes.
- Allows plug and play solution (only requires energy and internet access).
- Low cost solution without need for reconstruction the home environment.
- Mobility: it will allow mobile videoconferencing, alarm evaluation, remote control by family members or social care services.

2. ASTERISK AND SIP CLIENTS ARCHITECTURE

Asterisk can be thought of as “middleware”, it sits between telephony technologies, applications and providing a generic interface between them. In other words, it connects to various telephony technologies by giving a consistent way to receive and send calls to and from various VoIP protocols, as well as to and from Integrated Service Digital Network (ISDN) and legacy Public Switched Telephone Networks (PSTN). At the same time, it gives a consistent to various telephony applications such as voicemail, conferencing, Interactive Voice & Video Response (IVVR) scripting and etcetera. The means for transporting a VoIP connection generally engage a sequence of signaling transactions between endpoints, gateways and culminating into two persistent media streams (one for each direction) that carry the actual conversation and there are several VoIP protocols in existence to handle this, thus SIP (Session Initial Protocol) is one of the protocols. SIP is an application-layer signaling protocol (Spencer, 2007), that uses the well known port 5060 for communication and it can be transported using transport-layer protocols either UDP (User datagram Protocol) or TCP (Transmission Control Protocol). This protocol will establish, modify and terminate multimedia sessions such VoIP calls, unlike RTP (Real-time Transport Protocol) it does not transport media between endpoints and RTP is the one that is responsible for transmission of media like voice between end points.

In figure 1**,** is a schematic architecture representation of SIP clients and Asterisk PBX servers, consequently for communication process this architecture requires SIP user agents (SIP softphones like Ekiga), SIP registrar servers (databases containing location of users agents within a domain), SIP proxy servers (for accepting SIP session requests, forward them) and SIP redirect servers (directs SIP sessions invitation to external domains).

Copy Rights

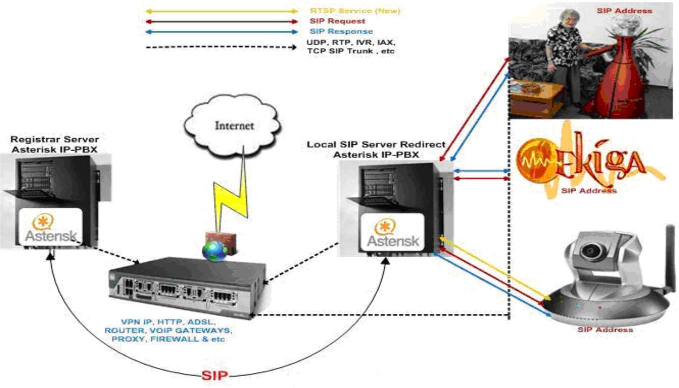

RTSP Service (New)
SIP Request
SIP Response
UDP, RTP, IVR, IAX, TCP SIP Trunk , etc
SIP Address
Internet
Registrar Server Asterisk IP-PBX
Asterisk
Local SIP Server Redirect Asterisk IP-PBX
Asterisk
Ekiga
SIP Address
VPN IP, HTTP, ADSL, ROUTER, VOIP GATEWAYS, PROXY, FIREWALL & etc
SIP Address
SIP

Figure 1. SIP Clients and SIP Servers

Each patient is equipped with a SIP softphone like Ekiga together with webcams and the robot machine will have its own equipment and will have its own SIP ID (Rocaries and Simonnet, 2008).The SIP client add some additional advantages on this project, it will be able to dynamically balance the compression ratio and the quality of both channels (audio and video) to adjust to different circumstances and will permit load balancing of bandwidth between sound and video streams.

2.1 Remote Control Prototype and Future Enhancement

Through the use of Glade tool, it is achievable to build a user friendly GUI of Ekiga for operators. For such developments, we used the latest version of Glade to design the Remote Control Prototype and we conducted the preliminary tests by creating and adding new Ekiga tab - Robot Control Tab - using C, C++ tool and by editing the XML codes with text editor. Consequently it is be possible to implement a Graphical GUI to pilot a robot through the use of keyboard that can be replaced in future by a joystick (see figure 2. for a proposed Remote Control Tab Prototype).

This project uses a bidirectional communication channel for the robot remote control. The SIP MESSAGE extension (RFC 3428) is used. The original use of this method is Instant Messaging (IM), and then can be used both for remote control but also for sending back sensors values and alarms. Then every sensor values are sent back through Asterisk using the SIP protocol. MESSAGE is handled by some softphones (Ekiga, LinPhone), but is not handled by Asterisk PBX. We developed a specific Asterisk module to handle MESSAGE on a central PBX but also to propagate IM to all PBX used for a specific call. It is necessary to have this function to bypass private IP addressing using a local PBX and trunk functionality.

We also developed an Asterisk module handle Julius ASR (Automatic Speech Recognition). This will allow automatic calls and orders recognition.

Copy Rights

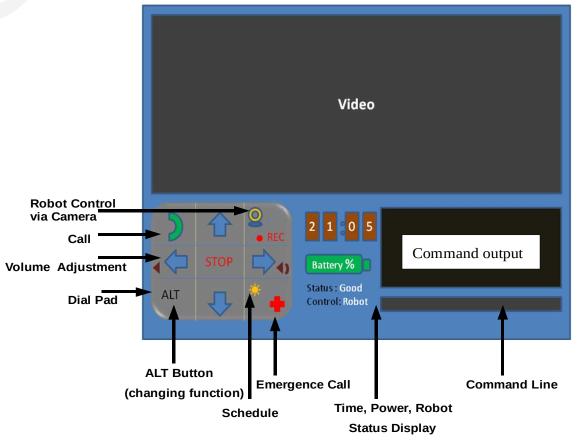


Figure 2. Remote control tab

3. CONCLUSION

Certain health services can be greatly enhanced by means of telemedicine technologies, for instance home health services are receiving a great deal of attention and investments in some areas. These technologies enable home health providers to redefine patient treatment plans, as they are able to increase patient visits due to elimination of significant percentage of travel to patients home. A home-care and community based health services are becoming an increasingly important part for old age people like patients affected by pre-Alzheimer disease (Hanett, 2006). There are many reasons for this including: patients are leaving hospital sooner because of medical expenses and they will need some additional care at their home premises while they recover, therefore treating patients at home is less expensive than treating them in the hospital. Consequence many patients prefer to stay in their home premises as long as possible before moving onto a higher level of healthcare service centers like nursing homes. As a result, the progress in technologies and availability of OpenSource tools offer the integration possibilities to control robot machines and allow bidirectional communication for videoconferencing support. This comes from the second successful annual review of CompanionAble project - http://www.companionable.net/. Through the integration procedures of a robot and a smart house, CompanionAble presents a care environment that supports family members and therapists in their daily task therefore a necessity to remotely control a robot machine is desirable. Such projects are important in our communities because they review the existing VoIP tools or attempt to find the best corresponding tools in order to promote flexibility of communication between medical professionals and medical doctors with their families especially in a position of trust relationship.

ACKNOWLEDGEMENT

I would like to thank F’SATIE collaborating with TUT and French Embassy for selecting me and financed my Masters academic year in France, not forgetting ESIEE Paris which allowed me to utilize its facilities and equipment for the project. As a result I have attained both practical and theory knowledge.

Some research leading to these results has received funding from European Community's seventh Framework Programme (FP7/2007-2013, under grant agreement n°216487), ANR - French National Research Association through QuoVADis project.

Copy Rights

REFERENCES

Bley A.. et al, 2008. Integrating Service Robots into ICT solution for technology supporting Ageing. *In Seventh Framework Programme on CompanionAble Project: FP7 Grant Agrrement Nr. 216487*. Berlin, Germany.

Boudy J. et al, 2006. Telemedicine for elderly patient at home: TelePat project. *International Conference on Smart Homes and Health Telematics.* Belfast, UK, pp. 1-8.

Callegari C., Pagano M., 2009. Security and Delays issues in SIP Systems. *Wiley InterScience International Journal on Communication Systems*, Vol. 22, No. 5,pp 1023-1044.

Clinton G., Mativo M., Thai C., 2009. Robotics-based Curriculum Development for an Immigration Course into Computer System Engineering. *In IEEE Xplore Southeastcon on Humanities, Social Science and Law,* Vol. 5, No. 6,pp 278-283.

Commission of the European Communities, 2009. The Health Report of European initiative on Alzheimer’s disease and other dementias. Brussels N° 380.

Couet A., Ezvan P., Givernaud O., Hillereau P., Simonnet Th., 2009. Telemedicine Platform Enhanced visiophony to operate a Robot-Companion. *In SpringerLink on Computer Science,* pp 301-305.

Dong L., Robert G., 2008. IMS Shared Streaming Video. *In Wiley InterScience on Bell Labs Technical*, Vol. 10, No. 4,pp 71-75.

Frandina S., 2010. Videoconferencing for Asterisk: Study and Implementation. Masters of Technology: in Telecommunication Engineering, University of Siena. Italy, March 2010.

Gharpure C.P., Kulyukin V. A., 2008. Robot-assisted shopping for the blind: issues in spatial cognition and product selection. *In Springer Netherlands SpringerLink on Intelligent Service Robotics and Engineering,* Vol. 1, No. 3,pp 237-251

Medjahed H., François Steenkeste, Andreao R. 2008. A Multimodal Platform for Database Recording and Elderly People Monitoring. *International Conference on Bio-inspired Systems and Signal Processing*. Setùbal, Portugal, pp. 385-392.

Hanett, B. et al, 2006. Telemedicine and telecommunication. International Research Development Centre RSM Publisher, London. UK.

Ko. Albert W. Y., Lau Henry Y. K., 2009. Intelligent Robot-assisted Humanitarian Search and Rescue System. *In Advanced Robotic Systems on Health Management Technology*, Vol. 6, No. 2,pp 121-128.

Madsen L., Meggelen J., Spencer M., et al, 2007. *Asterisk: The Future of Telephony*. O'Reilly Media Inc Publisher. Gravenstein Highway North, Sebastopol. Heidelberg, Germany

Mark., 2006. *Introducing Gtk+: Glade and User Interface.* Apress Publisher. San Francisco. USA.

Mupparapu M. et al, 2008. VoIP for Orthodontic practice: A sensible switch from plain old telephone service. *In American Journal of Orthodontics and Dentofacial Orthopedics on Information and Knowledge Engineering*, Vol. 113, No. 3,pp 133-470.

Pavlidou F., Stalianos K., 2009. VoIP a comprehensive survey on a promising technology. *In SscienceDirect on Compunter Networks*, Vol. 4, No. 53,pp 2050-2090.

Rocaries F., Simonnet Th., 2008. Collaborative tools for a Telemedicine Platform: Monitoring, Communication and Storage. *In IADIS International Journal on Informatics,* pp 193-198.

Seeling P. et al, 2010. Scene Change Detection for Uncompressed Video. *In SpringerLink Netherlands White Paper on Humanities, Social Science and Law*, Vol. 4, No. 1,pp 11-14.

Spencer M. et al, 2007. Voice over IP: understanding the basic network functions, components and signaling protocols in VoIP networks. *In Juniper Networks White Paper on Computer Systems*, Vol. 2, No. 3,pp 1-21.

Simonnet T., 2009, Telemedicine Platform Enhanced Videoconferencing solution to operate a Robot-Companion. *IADIS International Conference on Informatics*. Algarve, Portugal.

Copy Rights

COMPARISON OF DISCRETIZATION METHODS OF FLEXIBLE-RECEPTOR DOCKING DATA FOR ANALYSES BY DECISION TREES

Karina S. Machado, Ana T. Winck, Duncan D. Ruiz and Osmar Norberto de Souza
LABIO - Laboratório de Bioinformática, Modelagem e Simulação de Biossistemas
GPIN - Grupo de Pesquisa em Inteligência de Negócio
PPGCC, Faculdade de Informática, PUCRS
Av. Ipiranga, 6681 – Prédio 32, sala 608, 90619-900, Porto Alegre, RS, Brazil

ABSTRACT

Molecular docking is an important step of the Rational Drug Design process where it is predicted the preferred conformation and orientation of a small molecule (ligand), to a biological macromolecule (receptor) in order to form a stable complex. To incorporate the receptor flexibility in docking we perform a series of docking simulations using in each one a receptor conformation constructed from a molecular dynamics simulation trajectory. This kind of approach generates vast amounts of data and is very computer-intensive. Aiming at reducing this computational demand and discovering information about the receptor-ligand interactions we apply the J48 implementation of C4.5 classification decision tree algorithm where the input mining files are based on the free energy of binding (FEB) that estimate the affinity of the receptor-ligand complex and distances between the receptor residues and the ligands. Since FEB is a continuous value and, in classification tasks, the target attribute must be categorical, we propose to compare three discretization methods for FEB: by equal frequency, by equal width and by mode and standard deviation and evaluate their impact in the generated decision trees. Moreover, using the induced decision trees we introduce a different approach to analyze receptor-ligand interactions from docking simulation results.

KEYWORDS

Classification; Decision Trees; Discretization; Bioinformatics; Rational drug design; Molecular docking.

Copy Right

1. INTRODUCTION

The use of sophisticated instruments and the massive use of computers has made molecular biology a science where the size of the data to be analyzed is a computational challenge (Chen and Stefano, 2009). These large biological data sets must be analyzed and interpreted to extract relevant information. Such analysis can be better performed and the results can be more useful if data mining techniques are applied.

We have mining biological data inserted into the context of Rational Drug Design (RDD) (Kuntz, 1992). One of the most important step of RDD is molecular docking, a computational method applied to predict the preferred conformation and orientation of a ligand to a second molecule, named target receptor, in order to form a stable complex (Lengauer and Rarey, 1996).

In molecular docking algorithms, usually the ligand is treated as flexible because it has few atoms. The limitation of these algorithms is in considering the flexibility of the receptors: molecules with hundreds or thousands of atoms (Totrov and Abagyan, 2008). Currently there are different approaches to incorporate the receptor flexibility in docking in which we chose to perform a series of docking simulations using in each one a receptor conformation produced by a molecular dynamics (MD) (van Gunsteren and Berendsen, 1990) simulation trajectory (Lin et al. 2002). This kind of approach to consider receptor flexibility in docking simulations generates vast amounts of data and is a very computer-intensive (Machado et al. 2007). However the docking experiments are more realist considering the receptor flexibility since proteins are inherently flexible systems and its can be determinant to find new drug candidates for target receptor proteins.

Aiming at, in the future, reducing the computational demand of these docking experiments and to better understand the importance of receptor flexibility we opted to apply a classification decision tree algorithm on the docking results. According to Freitas et al. (2010) a decision tree output has the advantage to graphically represent the discovered knowledge and to point out to the importance of the attributes used for prediction. Since we are working in an interdisciplinary area, we need to choose a classification algorithm where the output has to be easily understandable and not a black box like support vector machines or neural networks outputs. To achieve this, we apply the C4.5 (Quinlan, 1986) classification decision tree algorithm (WEKA J48 implementation). From all the docking results we defined as predictive attributes the distances between the receptor residues and the ligands, and, as target attribute, the estimated free energy of binding (FEB). FEB is a physics-based scoring function that estimates the affinity of the receptor-ligand complex after the docking simulation. Since FEB is a continuous value and, in classification tasks, the target attribute must be categorical (Tan et al. 2006), we need to discretize the FEB value.

In this work we propose to compare different discretization methods (by equal frequency, by equal width and by mode and standard deviation) and evaluate their impact in the induced trees, analyzing which is the most promising discretization method for this kind of biological data. Moreover, using the induced trees we introduce a different approach to analyze receptor-ligand interactions from docking simulation results.

2. MATERIAL AND METHODS

In this work we are considering as receptor the InhA enzyme from *Mycobacterium Tuberculosis* (Mtb) (Dessen et al. 1995). As ligands we used the pentacyano(isoniazid)ferrate(II) (PIF) (Oliveira et al, 2004), nicotinamide adenine dinucleotide (NADH) (Dessen et al. 1995), triclosan (TCL) (Kuo et al. 2003) and ethionamide (ETH) (Banerjee et al, 1994).

Among all the different approaches to consider the receptor flexibility in docking, reviewed by Totrov and Abagyan (2008) we chose to perform a series of docking simulations considering in each one a different receptor snapshot generated by a molecular dynamics (MD) simulation (van Gunsteren and Berendsen, 1990). In doing so, starting from the crystal structure of this receptor (PDB ID: 1ENY), by means of MD simulations we generated the flexible-receptor model of InhA. It is made up of 3,100 snapshots derived from a 3.1 ns (1 ns = 10^{-9} seconds) MD simulation trajectory (Schroeder et al., 2005). The docking simulations were executed with AutoDock3.0.5 (Morris et al. 1998) considering the flexible InhA receptor model and each of the four ligands described above (Machado et al. 2007). The data containing the MD simulation trajectory snapshots and the related docking results were stored in the FReDD repository (Winck et al. 2009).

We are performing data mining applying the J48 algorithm available on WEKA (Hall et al. 2009). J48 is an implementation of the decision tree classification algorithm C4.5 (Quinlan, 1986). Our input data for each ligand is composed by instances that correspond to each docking result stored in the FReDD (Winck et al. 2009) repository. For each instance the predictive attributes are the minimum distances between the 268 receptor residues and the ligand (measured in ångströms (Å)), and the target attribute is the best FEB value for each of the 3,100 docking simulations. Since classification decision trees require a categorical target attribute, and being FEB a continuous one, part of the data preparation step needed to include the discretization (Tan et al. 2006) of the distribution of FEB values.

2.1 Discretizing the Target Attribute FEB

In this work we consider three unsupervised discretization methods of FEB (Machado et al. (2010)):

- Method 1-by equal frequency: being *k* the number of intervals and *m* the total number of instances, the continuous variable is divided into *k* intervals, where each interval contains *m/k* values.
- Method 2-by equal width: the continuous attribute is sorted and divided into *k* intervals where each interval has the same width (Tan et al. 2006).
- Method 3-by mode and standard deviation: This method proposed in Machado et al. (2010) divide the sorted attribute into intervals considering the mode and standard deviation of the frequency distribution of the attribute that is being discretizing.

The three discretization methods were applied to our target attribute FEB. As a result we mapped the FEB values into 5 classes: *Excellent*, *Good*, *Regular*, *Bad* and *Very bad*. The results of the docking simulations and discretization methods are presented on Table 1 where the columns 6-10 display the total number of instances in each FEB class according to the discretization methods.

Table 1. Docking simulations results and mapping of instances into FEB classes according with discretization methods. Column 1 has the ligand names. Column 2 contains the total number of valid docking results, columns 3 the average and standard deviation of estimated FEB (in kcal/mol) and column 4 the mode value of FEB distribution. Columns 6-10 display the total number of instances for each of the 5 FEB classes considering the three different discretization methods.

Ligands	Dockings	FEB	Mode	Method	Excellent	Good	Regular	Bad	Very Bad
PIF	3,042	-9.9 ± 0.6	-9.9	1	604	607	620	610	601
				2	299	26	17	3	1
				3	7	223	2616	173	23
NADH	2,823	-12.9 ± 4.2	-16.8	1	569	559	565	565	565
				2	757	792	839	408	27
				3	205	1020	374	903	321
TCL	2,837	-8.9 ± 0.3	-9.0	1	563	556	587	582	549
				2	1017	1814	4	0	2
				3	19	158	1866	645	149
ETH	3,043	-6.8 ± 0.3	-6.7	1	619	591	598	649	586
				2	18	173	1108	1531	213
				3	160	512	2131	226	14

2.2 Applying the J48 Algorithm

We generated input files for the four ligands and for each of the three discretization methods. We executed J48 in WEKA with the parameter related to the number of instances in each leaf node set to 50 (this set up should allow the generation of more legible trees). The other parameters remained with default values.

For decision trees there are some typical measures: accuracy (Acc) is the rate of instances that were correctly classified; tree size (TS) concerns with the number of nodes in the generated tree: the smaller the tree size, the better the model (more interpretable trees); root mean-squared error (RMSE) and mean absolute error (MAE) make use of the predicted values *p_1 … p_n* and the actual values *a_1 … a_n*. The F-measure (FM) is the rate that considers precision and recall. Smaller values of RMSE and MAE and higher values of FM correspond to better mining results. In addition, we evaluate the rate of instances that belongs to the Excellent or Good classes (IEGC), where we are looking for the smaller rates.

Copy Rights

3. RESULTS AND DISCUSSION

The results of the twelve J48 executions are summarized in Table 2. Each execution corresponds to one line on the table and each column shows the performance measures for the generated decision-tree models.

Table 2. Results of J48 executions. The discretization methods and ligands are shown ih the first two columns. Columns 3 to 7 contains the decision-tree measures. Highlighted on the table are the best values of the measures by ligand.

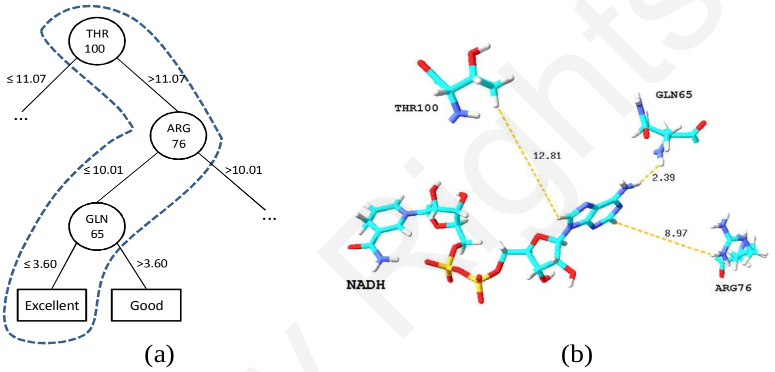
Method	Ligand	Acc	TS	MAE	RMSE	FM	IEGC
1	PIF	31.92	71	0.30	0.40	0.31	39.81
	NADH	61.88	61	0.18	0.32	0.62	39.96
	TCL	30.49	61	0.30	0.40	0.30	39.44
	ETH	33.37	77	0.28	0.39	0.35	39.76
2	PIF	98.68	3	0.01	0.07	0.98	99.31
	NADH	73.53	43	0.14	0.28	0.73	54.87
	TCL	64.93	49	0.16	0.30	0.64	99.79
	ETH	61.02	41	0.21	0.33	0.57	06.28
3	PIF	86.55	5	0.09	0.22	0.81	07.56
	NADH	75.41	35	0.13	0.27	0.75	43.39
	TCL	66.23	17	0.19	0.31	0.58	06.06
	ETH	70.32	29	0.17	0.29	0.65	22.08

Method 1, by equal frequency, had the worst results for all ligands, showing that this kind of discretization is not a good option to apply on the docking results.

Method 2, by equal width, obtained better performance measures for PIF. However, for this ligand, this method has as 99.31% of the instances into the class Excellent or Good (Table 2, IEGC). It means that the generated model for PIF-Method 2 is not useful to extract information about receptor residues involved in good docking results because almost all instances fitted into the same classification category. For TCL, Method 2 was better in three of the five performance measures. But, as for PIF, 99.79% of the instances are previously classified as Excellent or Good. If we look at IEGC values, for PIF and TCL, we can observe that although the accuracy is better, the generated models are distorted.

Method 3 proposed on Machado et al. (2010), obtained the best performance measures for ETH and NADH and two of five for TCL. Although the performance measures were not the best for all ligands, the generated decision trees using this discretization method were more legible (with smaller size). Consequently they are more useful. Thus, with these decisions trees, we are able to extract information about the relationship between the flexible receptor residues and the FEB classes.

It is beyond the scope of this work to discuss in detail the generated decision trees. However, as an example of useful information that can be extracted from these trees, we show in Figure 1a part of the NADH-Method 3 decision tree, where the path to the class *Excellent* is highlighted. Figure 1b shows instances of minimum distances of the NADH ligand to the flexible receptor residues shown in Figure 1a.



(a) (b)

Figure 1. (a) Part of the NADH-Method 3 decision tree. (b) The three flexible InhA receptor residues used by the NADH-Method 3 decision tree to achieve the Excellent FEB class. The distances values are in Å.

In Figure 1a, with the NADH-Method 3 decision tree, we can notice that the docking results classified as Excellent FEB always follow the rules: the receptor residue Threonine 100 (THR100) has to be at a distance bigger than 11.07 Å from NADH, Arginine 76 (ARG76) must be in a distance smaller than 10.01 Å from NADH, and Glutamine 65 (GLN65) has to be at a maximum distance of 3.60 Å from NADH.

Similar analyses can be done with all generated Method 3 decision trees. In our point of view, this kind of investigation, using the results of decision trees, is a new approach to explore flexible receptor-ligand interactions based on their corresponding docking results.

4. CONCLUSION

We previously performed docking simulations with a flexible-receptor model of the InhA enzyme from Mtb and four ligands: PIF, NADH, TCL and ETH. We performed J48 executions with the generated input files for the four ligands and the three discretization methods. The results of performance measures from J48 executions show that the method of discretization by equal frequency is not satisfactory. Although the method by equal width has good measures for two of the four ligands, the discretization was unbalanced and did not generate good decision trees. The method by mode and standard deviation is better in all performance measures for two ligands and the obtained decision trees for all ligands are the more legible. For these reasons we believe that Method 3 is the best discretization method for this type of data. This method can be

Copy Rights

applied to other biological data that has a continuous target attribute that needs to be discretized and where the generated trees should provide relevant information for the decision makers.

Finally, in this work we have shown a new approach to analyze receptor-ligand interactions based on the obtained decision trees. As future work we plan to analyze in detail the obtained decision trees to establish characteristics in the snapshots that obtained good docking results aiming at selecting the most promising snapshots and, consequently, diminishing the computational demand to perform flexible-receptor docking simulations routinely.

ACKNOWLEDGEMENT

This work was supported by grants from MCT/CNPq 14/2008 to DDR and in part by grants 410505/2006-4 and 312027/2006-0 to ONS. ONS is a CNPq Research Fellow. ATW is supported by CT-INFO/CNPq 17/2007 PhD scholarship. KSM is supported by a CAPES PhD scholarship.

REFERENCES

Banerjee, A., et al. 1994. InhA, a gene encoding a target for isoniazid and ethionamide in Mycobacterium tuberculosis. *Science* Vol. 263, pp. 227- 230.

Chen, J.Y. and Stefano, L. 2009. *Biological Data Mining*. Chapman & Hall, New York, USA.

Dessen, A., et al. 1995. Crystal structure and function of the isoniazid target of Mycobacterium tuberculosis. *Science* Vol. 267, pp. 1638-1641.

Freitas, A., Wieser, D. and Apweiler, R. 2010. On the importance of comprehensible classification models for protein function prediction. *IEEE/ACM Trans. Comput. Biol. Bioinformatics* Vol. 7, pp. 172-182.

Hall, M., et al. 2009. The WEKA data mining software: an update. *SIGKDD Explorations* Vol. 11, pp. 10-18.

Kuntz, I. 1992. Structure-based strategies for drug design and discovery. *Science* Vol. 257, pp. 1078-1082.

Kuo, M. et al. 2003. Targeting tuberculosis and malaria through inhibition of Enoyl reductase: compound activity and structural data. *J. Biol. Chem.* Vol. 278, pp. 20851-20859.

Lengauer, T. and Rarey, M., 1996. Computational methods for biomolecular docking. *Curr. Opin. Struct. Biol.* Vol. 6, pp 402-406

Lin, J.-H., et al. 2002. Computational drug design accommodating receptor flexibility: the relaxed complex scheme. *J. Am. Chem. Soc.* Vol. 124, pp. 5632-5633.

Machado, K. S., et al. 2007. Automating Molecular Docking with Explicit Receptor Flexibility Using Scientific Workflows. *Lect. Notes Comput. Sci.* Vol. 4643, pp. 1-11.

Machado, K.S. et al. 2010. Discretization of Flexible-Receptor Docking Data. *Lect. Notes Comput. Sci.*, Vol. 6268, pp. 75-79.

Morris, G., et al. 1998. Automated docking using a Lamarckian genetic algorithm and an empirical binding free energy function. *J. Comput. Chem.* Vol. 19, pp 1639-1662.

Oliveira, J., et al. 2004. An inorganic iron complex that inhibits wild-type and an isoniazid-resistant mutant 2-trans-enoyl-ACP (CoA) reductase from Mycobacterium tuberculosis. *Chem. Commun.* Vol. 3, pp. 312-313.

Quinlan, J. 1986. Induction of Decision Trees. *Mach. Learn.* Vol. 1, pp. 81-106.

Schroeder, E., et al. 2005. Molecular dynamics simulation studies of the wild-type, I21V, and I16T mutants of isoniazid-resistant Mycobacterium tuberculosis enoyl reductase (InhA) in complex with NADH: toward the understanding of NADH-InhA different affinities. *Biophys. J.* Vol. 89, pp. 876-884.

Tan, P., Steinbach, M. and Kumar, V. 2006. *Introduction to data mining*. Addison Wesley, Boston, USA.

Totrov, M. and Abagyan, R. 2008. Flexible ligand docking to multiple receptor conformations: a pratical alternative. *Curr. Opin. Struct. Biol.* Vol. 18, pp 178-184.

van Gunsteren W. and Berendsen, H. 1990. Computer simulation of molecular dynamics: methodology, applications, and perspectives in chemistry. *Angew. Chem. In.t Ed. Engl.* Vol. 29, pp. 992-1023.

Winck, A., et al. 2009. FReDD: supporting mining strategies through a flexible receptor docking database. *Lect. Notes. Comput. Sci.* Vol. 5676, pp. 143-146.

COMPONENT INTERCONNECTION INFERENCE TOOL SUPPORTING THE DESIGN OF SMALL FPGA-BASED EMBEDDED SYSTEMS

Zbyněk Křivka, Ota Jirák and Zdeněk Vašíček
Faculty of Information Technology, Brno University of Technology
Božetěchova 2, 612 66 Brno, Czech Republic

ABSTRACT

The paper studies use of component technology to simplify the design of small FPGA-based embedded system. Based on the structural part of WRIGHT component model, we introduce a tool with a knowledge base supporting the creation of a configuration using the given architecture. The design of the tool includes an algorithm for the semiautomatic inference of a component interconnection defined by a user on the higher level of abstraction. We present a demonstration example of a tool application used in the educational process.

KEYWORDS

Architecture, knowledge base, interface, components, ports, connectors.

1. INTRODUCTION

There is a system for the support of an embedded system design at the educational level (bachelor and master degree) developed at our faculty. Students often complain about the problematic accessibility of hardware devices such as evaluation and demonstration boards and peripherals apart from the scheduled laboratory times. Therefore, we are building virtual laboratory with remote access to handle these students' request. The laboratory is focused on the microprocessor and programmable hardware (FPGA) technologies. One of the principal goals is to develop an integrated development environment consisting of several supporting tools for the given hardware considering the necessity of the remote access.

This paper introduces a design of the tool supporting component-based design for a given hardware architecture using the corresponding knowledge base. Here, we describe the design of such tool including the knowledge base representation, configuration representation, and semiautomatic interconnection inference. As a framework for the representations, we use the structural part of WRIGHT component model (see [Allen, 1996] for the details).

Although there exist commercial tools that deals with the similar problem (i.e. component-based design), all the available tools are based on a simplified approach of this problem. For example, Xilinx EDK provided by Xilinx [Xilinx] and Altium Designer [Altium] are the tools designed for rapid application development of FPGA-based systems. Both of these tools are based on the same approach—each tool is equipped with a database of unified components (i.e. the components with a predefined interface). While Xilinx EDK uses PLB and OPB bus designed by Xilinx, Altium Designer uses a Wishbone (open source bus). Since the signals of each component that are used for the interconnection have to match the signals of a supported interface, the interconnection of the components is straightforward, e.g. it can be based on matching of signal names only. The main disadvantage of this approach is the limitation to a supported interfaces only, thus these systems can not be used for interconnection of arbitrary components which is our goal. Apart from this, there exists another approach. For example, PSoC Creator [CYPRESS] is a tool designed for development of embedded systems on a chip. The design process is based on the individual signal connection by means of wires (i.e. the user is responsible for the proper interconnection of the components). While this approach offers higher flexibility in comparison with the previous one, it requires higher user's proficiency because the user has to know what is possible to interconnect.

Copy Rights

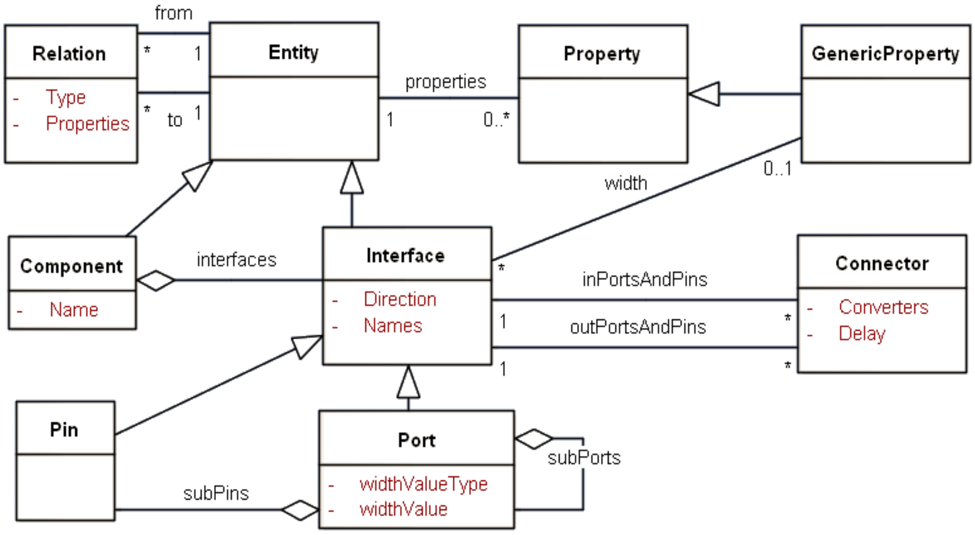


Figure 1. Core of class diagram in UML to model configurations

The following examples and the proposed tool itself are primarily focused on FITkit demonstration board developed at our faculty. FITkit (see [FITkit]) is a platform integrating 16-bit microcontroller (MCU) of MSP430 family, small reprogrammable FPGA and several peripherals such as LCD display, VGA interface, SPI interface, etc. The main advantage of this platform is that MCU can solve complex but not time-consuming tasks while FPGA can provide the power for the data-demanding and time-consuming part of the computation such as producing of graphical output, Ethernet communication, etc. This concept gives students opportunity to learn HW/SW co-design techniques. Moreover, software development tools such as MSP430GCC compiler and Xilinx ISE Webpack for small FPGAs (see [Xilinx]) are available for free. FITkit communicates with personal computer through USB connection (i.e. it does not need any special programmer).

2. DESIGN OF THE SUPPORTING TOOL

The application is based on visual editor generators in Eclipse Modeling and Graphical Modeling Frameworks (EMF, GMF) and will be part of the Eclipse-based integrated development environment (see [Jirák, 2010]). The tool that helps students to develop a simple embedded system design using the laboratory demonstration hardware (focused on small FPGAs) by an intuitive graphical interface contains three parts:

1. *Knowledge base and Configuration* – the storage for facts and relations between hardware components necessary for the interconnection inference with the support for the code generation for the following synthesis.
2. *Inference support* – the procedure provides semiautomatic components interconnection inference based on the analysis of several basic properties such as the signal width, direction, and type; it is inspired by Prolog unification and a tree matching. The goal of our new algorithm is to support the user by the list of advices which components (from the chosen collection) to interconnect and which ports are preferred to connect to each other.
3. *Graphical user interface* – as a tool for students, we must provide an intuitive user interface, too.

As other parts the tool will cooperate with other existing tools such as compilers, downloading tools, and synthesizers.

Copy Rights

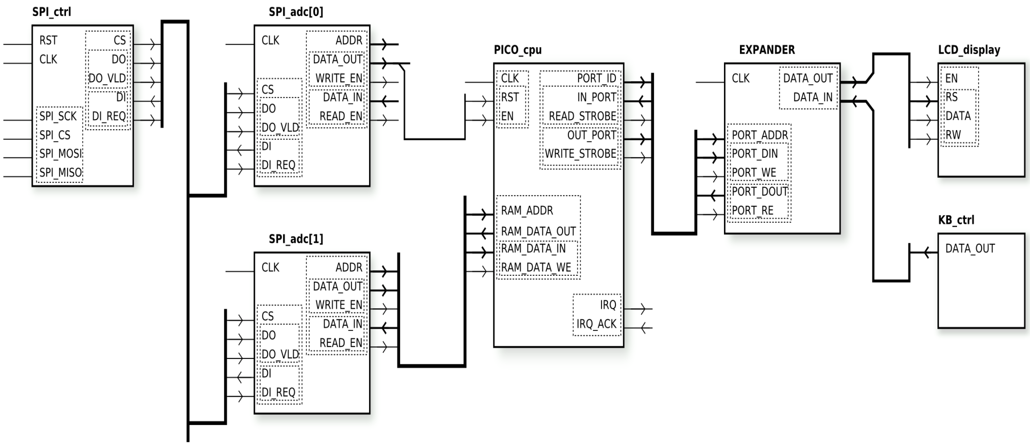


Figure 2. Configuration example of SPI-based architecture at FITkit platform

2.1 Knowledge base and Architecture Configuration

The *knowledge base* describes several similar embedded system architectures for the given demonstration board and hardware devices where every system entity (component, connector, port) is defined by a static collection of attributes and relations such as name, alternative names, properties, constrains, and initial settings.

The *configuration* of the system architecture specifies particular however static interconnection of component instances (see the reduced class diagram in Figure 1 omitting many less interesting attributes, enumerations, and classes). In fact, Figure 1 demonstrates the static core model of configurations. Figure 1 without classes *Connector* and *Relation* shows also the core of knowledge base model. This allows us to unify the processing of common configuration and knowledge base elements.

Let us explore the diagram in more details. *Interface* is a generalization of *Port* and *Pin* classes that represent a hierarchical structure and a unit for the interface of every component (*Component*) inspired by Composite design pattern, respectively. Every interface has defined or derived its width and signal direction that are the most essential attributes of interconnection inference. More specifically, *widthValueType* attribute determines how port width is computed. If its type is *direct*, *widthValue* attribute is valid and containing a direct value; if the type is *inferred*, the port width is inferred from its subports and subpins width; otherwise, for *generic* type, *width* relation refers to *GenericProperty* object containing the constant value (e.g. address width, data width). *Names* is an unconventional attribute containing an ordered list of names (the most specific name comes first) of the interface to guess the interconnection mapping between the similar interfaces (with respect to the width and direction). Every port that is a leaf of the interface hierarchy stands for a list of corresponding number of pins.

We use *Connector* to fully or partially interconnect component interfaces to model a communication from the source component (interfaces, *inPortsAndPins*) to the target component (interfaces, *outPortAndPins*). In general, the connector can bound several interfaces on both sides to represent scattered and/or partial interconnection. The connection can be augmented by the ordered list of converters (*Converters*) to adjust the signal to the special parameters of the interconnected interfaces such as the different logical activation level of a signal (negation converter) or the introduction of a required latency (delay converter).

The problematic interconnections that cannot be satisfied correctly by the inference algorithm are forced or forbidden by additional constrains in *Relation* with *forbidding-relation* or *forced-relation* value of *Type* attribute. For better flexibility, *Property* class supports arbitrary additional properties for each *Entity*.

Moreover, in some properties of the knowledge base or configurations, the standard value domains are augmented by value types such as valid value (*direct* or *set*), undefined value (*unset* or *missing*), future user-defined value (*generic*), inferred value (*inferred*), and free value (*unbound*).

Copy Rights

2.2 Inference Procedure

Notice from the graph theory that the problem of unordered tree matching that corresponds to the interconnection inference problem is NP-complete (see [Kilpeläinen, 1992]) and therefore, the procedure suits small designs only or requires user interaction. Specifically, we are not trying to solve every logical and semantic dependency as some architecture description languages try to do (see [Zivojnovic, 1996] and [Leupers, 1998]).

Interface Compatibility. For two given interfaces, we check their compatibility according to the capacity (how many connectors can be connected to this interface) and occupancy, width in bits and direction (in, out, in/out, mixed), and type (address, data, synchronization/clock, control). The procedure returns the *measure of compatibility* that equals the number of pins that can be interconnected without a violation of any constraint (see *Relation* in Figure 1 and the end of the previous section).

Matching interfaces. We start the matching at the highest level of the interface abstraction and continue by descending into the ports hierarchy, if necessary. The interface compatibility test starts at the source component and tries to fully match a subset of interfaces in the target component with respect to the interface compatibility and maximal measure of this compatibility. As this matching is highly ambiguous, we use the list of names of examined ports to validate probably correct connections. Before a change of the source port name, every alternative name of the target interface is examined into given depth difference. If two interfaces do not match, try any other combination on the same level of the hierarchy (analogical with the breath-first search); otherwise, choose more width interface (source or target) and descend into its inner interfaces and repeat the matching.

Interconnection inference. Globally, we are searching for the greatest compatibility measure that, in addition, matches two subsets of interfaces between two components interconnected by user who specifies source and target component. In fact, the source component is the master component in the connection. The compatibility measure is influenced by several weighted parameters such as the width, the order of source and target interface name (see *Names* attribute in Figure 1), the number of required converters, etc. In general, the determination of the most suitable weights is an open problem. We focus on this issue in several experiments for FITkit platform right now.

As a result of this procedure, the user gets the list of possible interconnections. Some of these suggested connections can be manually marked as appropriate or unsuitable (temporarily set as forced or forbidden connections) and the inference procedure can be repeated with new additional information.

2.2.1 Example Explanation

In order to demonstrate proposed inference procedure, let us consider a simple design for FITkit platform consisting of PicoBlaze processor that controls LCD and keyboard (Figure 2). PicoBlaze can be programmed from MCU (omitted from Figure 2) through SPI bus controller (SPI_ctrl). The address decoders (SPI_adc[0/1]) are connected with SPI_ctrl. SPI_adc[0] controls PicoBlaze processor (PICO_cpu). EXPANDER expands ports from 8 to 16 bits. It enables the straightforward connection of LCD display and the matrix keyboard controller (KB_ctrl). SPI_adc[1] is used for PICO_cpu programming. The direction of each port (in bold) and pin is marked. The dashed lines indicate a port structure. As a simplification, the first name in a dashed block represents the name of the corresponding port or pin, e.g. CLK@PICO_cpu represents block with CLK pin and RST port containing RST and EN pins. The clock connections are omitted.

The inference algorithm is demonstrated on the typical aspects of interconnections in Figure 2:

1) The connection of SPI_adc[0] and SP_ctrl is obvious; SPI_adc[1] have to be analyzed for DI@SPI_ctrl pin accessibility (the semantic allows multiple sources as CS and DI_REQ pins make the chip selection). The incompatibility between CS@SPI_ctrl and ADDR@SPI_adc is due to widths and directions. The encapsulation of DO and DI ports prevents unwanted shuffle of pins.

2) An alternative name (e.g. "SPI_adc_DATA") and the algorithm rule to link the maximal width with an unoccupied port lead to the interconnection of RST@PICO_cpu and DATA_OUT@SPI_adc.

3) RAM_DATA_OUT@PICO_cpu port provides valid data every clock cycle, so READ_EN pin remains unconnected in DATA_IN@SPI_adc port.

4) Thanks to the straightforward matching of PORT_ID@PICO_cpu and PORT_ADDR@EXPANDER port, the interconnection between EXPANDER and PICO_cpu is simple. On the other hand, IRQ@PICO_cpu has no compatible name in the list of names.

Copy Rights

5) As LCD_display does not require a controller and as we choose a suitable alternative name of EN@LCD_display port (e.g. "EXPANDER_DATA"), the unambiguous connection between LCD_display and EXPANDER is possible. This connection leaves several unattached output pins.

6) There is no problem with KB_ctrl connection. An appropriate port (DATA_IN@EXPANDER) is free and it exactly correlates with DATA_OUT@KB_ctrl.

3. CONCLUSION

This paper proposes a tool supporting user-friendly designing of small FPGA-based systems. The knowledge base and the inference algorithm based on simplified WRIGHT component model introduce some problems of component-based design of embedded systems such as low-level dependencies violating the component abstraction and encapsulation. Hardware components are less autonomous in comparison with software components and therefore the code generation is another challenging task that is left for the future investigation and development. Although, there are similar tools, e.g. XILINX EDK (predefined components), PSoC Creator (signal interconnection), our tool investigates the combination of advantages of both approaches to provide flexible and robust solution.

ACKNOWLEDGEMENT

This work was supported by the Czech Ministry of Education, Youth and Sports grant MŠMT 2C06008 "Virtual Laboratory of Microprocessor Technology Application" (visit the website http://www.vlam.cz), the BUT FIT grant FIT-10-S-2, and the research plan MSM 0021630528.

Copy Rights

REFERENCES

Allen, D. et al., 1996. The WRIGHT architectural specification language (Technical report).

Altium Limited: Altium Designer. Available at: http://www.altium.com/products/altiumdesigner/.

CYPRESS Semiconductor Corporation: PSoC Creator. Available at: http://www.cypress.com/?rID=39551.

FITkit pages. Available at: http://merlin.fit.vutbr.cz/FITkit/.

Jirák, O. et al., 2010. Hardware Design Tool based on Eclipse Modeling Framework Model Specification. *Proceedings of the 44th Spring International Conference Modeling and Simulation Systems.* Ostrava, CZ: MARQ, pp. 138-144.

Kilpeläinen, P., 1992. Tree Matching Problems with Applications to Structured Text Databases (Dissertation thesis). University of Helsinki.

Leupers, R. et al., 1998. Retargetable code generation based on structural processor descriptions. *Des. Autom. Embedded Syst.*, Vol. 3, No. 1, pp. 75-108.

Xilinx, Inc.: Design Tools. Available at: http://www.xilinx.com/ISE.

Zivojnovic, V., et al., 1996. LISA - machine description language and generic machine model for HW/SW co-design. *IEEE Workshop on VLSI Signal Processing.* San Francisco, CA, pp. 127-136.

MINIATURE SIP FOR EMBEDDED SYSTEMS

Leonardo Maccari Rufino and Antônio Augusto Fröhlich
Federal University of Santa Catarina
Laboratory for Software and Hardware Integration
P.O.Box 476, 88040900 - Florianópolis - SC - Brazil

ABSTRACT

Session Initiation Protocol (SIP) is an application layer protocol responsible for creating, modifying, and terminating a session. It is used to establish calls through networks via IP protocol. Nowadays, SIP has been used by many embedded systems, however, due to its large size, some constrained systems cannot use it. This paper aims at adapting the protocol to be used in resource-constrained embedded systems, like household appliances and low cost surveillance cameras. In order to achieve this objective, many requests and header fields were suppressed, without compromising its functionalities, resulting in a reduced SIP. The reduced version was implemented in an embedded operating system, reaching a final system image of 116 Kbytes.

KEYWORDS

SIP, embedded systems, resource constraints.

Copy Rights

1. INTRODUCTION

SIP (Session Initiation Protocol) is an application layer protocol responsible for creating, modifying, and terminating a session (e.g. multimedia communication session such as voice and video calls) (Rosenberg et al. 2002). This protocol is used to establish calls through networks via IP protocol. Due to its characteristics, the SIP has been widely used in embedded systems, like cell phones, PDAs, and web cameras (Lakay and Agbinya 2005). However, the protocol lacks on performance (e.g. size of code) and cannot be used in resource-constrained embedded systems (e.g. household appliances and low cost surveillance cameras).

Some embedded systems do not have one person controlling it. These kinds of systems are the target of this work. An example is a security camera, where a session can be started either by a user making a call via a SIP phone for the address of the camera or by the camera itself, when motion is detected.

This paper proposes an adaption of the SIP protocol in order to enable its utilization in resource-constrained embedded systems. The proposed implementation of the protocol does not use some features, and requests and headers fields present in a full SIP version. Nevertheless, the modifications have not compromised the protocol functionality. The miniature SIP version was implemented using the Embedded Parallel Operating System (EPOS) (Fröhlich 2001), with a total of 116Kb of code and data.

The rest of this paper is organized as follow: Section 2 presents the related work. The proposed SIP implementation is described in Section 3. Section 4 shows the results. Finally, Section 5 concludes the paper.

2. RELATED WORK

Real-Time Visitor Communication Service (RVCS) is an architecture based on the home gateway system and SIP protocol, in which the resident of a home can monitor visitors in front of his/her door using devices connected to the Internet, such as PCs and PDAs, supporting user mobility (Oh et al. 2006).

SIP Context-Aware Gateway (SCAG) represents an intelligent gateway deployed between the home network and the Internet, allowing the homeowner to use his/her preferred SIP devices to communicate and post their context information for household appliances. It makes possible to smart home applications offer services that can be adapted to user’s dynamic situations (e.g. user location) and gives special treatment to their preferences (Cheng et al. 2006).

In addition, a design of a ubiquitous services system, which provides mobility and uniform interface to the user, using SIP, was proposed by Kwak (Kwak 2007). When a user wants to enter a SmartSpace (scenario which consists of intelligent services accessible to mobile users via handheld devices connected over wireless links for short distance), it registers its device in to a manager. Thus, the portable device receives the updated list of devices that exists in the SmartSpace and, after selecting the desired service and the destination device, the user is able to connect to it via a uniform interface based on touch event operations. Then, the requested service can be offered to the user.

The ability of processing SIP messages quickly is critical to the performance of consumer electronic devices in home network. Taking into account that the entire message parsing is the performance bottleneck of SIP servers, Demand-Driven Parsing Method (DPM) only parses the message types defined in a XML document, ignoring the others, thus enabling to save the processing time (Liao et al. 2009).

3. PROPOSED IMPLEMENTATION FOR THE SIP PROTOCOL

In order to obtain a miniature version of the SIP protocol able to run in a resource-constrained embedded system, without compromising its functionalities, only a subset of request and header fields were used. In the next subsections, we present the request and header fields that were implemented and discuss why some fields were not needed.

3.1 Requests

Among all the request messages that the SIP has, the implemented version is restricted to only three: *INVITE*, *ACK*, and *BYE*. Such messages represent the methods required for a session to be started and completed. Other requests such as those present in the original standard *CANCEL*, *OPTIONS*, and *REGISTER* (Rosenberg et al. 2002) and those located in separate RFCs, such as *REFER*, *SUBSCRIBE*, *NOTIFY*, *MESSAGE*, *UPDATE*, *INFO*, and *PRACK* (Roach 2002), have not been implemented. These kinds of requests are extensions of standard and therefore are optional. Consequently, they are not used in the SIP adaptation proposed in this paper.

Related to the SIP RFC requests, *CANCEL* is used to cancel a request previously sent by a UAC (User Agent Client). This message is not used because when an invitation to start the session comes from the embedded system, there is no need to cancel the attempt call. But when the attempt is made by the opposite side, as the system has no user, there is no need to send provisional responses (e.g. ringing) and the connection should be accepted or rejected instantly. Then, the *CANCEL* method can be suppressed without impact on functionality, because it has no effect on a request to which a UAS (User Agent Server) has already sent a final response, being applicable in situations where the server side can take a long time to respond to a request.

The *OPTIONS* method is used to query a UA (User Agent) or proxy server about their capabilities without the need to place a call with the other party. For example, before a UAC sends an *INVITE* message with the *Require* header field, it can send an *OPTIONS* to make sure that the UAS supports the required fields. This sequence of two messages, *OPTIONS* and *INVITE*, has the same effect of sending *INVITE* twice. In this situation, the first invitation requires a few options. However, if the called party does not support the required extensions, the standard procedure is to reject the request, sending a header field explaining the reason. Therefore, a second *INVITE* can be generated with only the options allowed by the UAS. Wherefore, this request can be removed without any problem.

REGISTER is a message used by a UA to notify a SIP network of its *Contact* URI and a URI which should have requests routed to this contact. This work suppressed this method because each UA has a fixed IP address. Then, the message exchanges are sent directly to the IPs of end-points desired, without the use of servers (proxy, redirect, and registrar).

3.2 Header Fields

The SIP protocol has 44 header fields described in its standard. Among these, only some have been used in this work: *Allow*, *Call-ID*, *Contact*, *Content-Disposition*, *Content-Length*, *Content-Type*, *CSeq*, *From*, *Max-Forwards*, *Record-Route*, *Require*, *Route*, *To*, *Unsupported*, and *Via*.

Copy Rights

Via, *From*, *To*, *CSeq*, *Call-ID*, *Max-Forwards*, and *Contact* are needed in the most of the SIP messages. *Content-Disposition*, *Content-Length*, and *Content-Type* identify the packets contained in the SIP message body. *Record-Route* and *Route* provide routing information. *Allow*, *Require*, and *Unsupported* report requests and header fields that may or may not be used.

All 29 remaining header fields were excluded. A major reason for the exclusion is that a portion of them only convey information about a user to another, however there is no need to exchange these data. Some headers that are included in this context are: *Accept-Language*, *Alert-Info*, *Call-Info*, *Content-Language*, *Date*, *Error-Info*, *Organization*, *Priority*, *Retry-After*, *Server*, *Subject*, *Timestamp*, *User-Agent*, and *Warning*.

Accept, *Accept-Encoding*, *Content-Encoding*, *MIME-Version*, *Proxy-Require*, and *Supported* were deleted from the miniature version because there is no available encoding for them, in other words, their fields would be empty if necessary send them. For instance, *Content-Encoding* indicates what additional content coding (e.g. gzip) have been applied to the message body.

This study does not require authentication schemes because it is designed for confined systems, not connected to the Internet. Consequently, some headers were excluded, as: *Authentication-Info*, *Authorization*, *Proxy-Authenticate*, *Proxy-Authorization*, and *WWW-Authenticate*. The remaining header fields were removed for their own reasons, for instance, *Expires* indicates the time that the message contents is valid, since after a response this field no longer has meaning, and as the responses are sent immediately on receiving a request, without having to wait for user actions, this header does not apply. *In-Reply-To* enumerates the Call-IDs that a call references or returns, *Min-Expires* transmits the minimum expiration time supported by a registrar server in response to a request *REGISTER*, as this work does not use this method, this header also becomes unnecessary. Finally, *Reply-To* specifies a URI that should be used in response to a request, however, as we are sending messages from fixed IP addresses, there is no need for a system has more than one URI identifying it.

The designed version has also the four state machines needed by the SIP protocol: *INVITE* client transaction, non-*INVITE* client transaction, *INVITE* server transaction, and non-*INVITE* server transaction. The miniature SIP version was implemented in the Embedded Parallel Operating System (EPOS) (Fröhlich 2001). EPOS was chosen because it has a small footprint and a complete communication stack, including UDP (used in the transport layer) and support for sensor networks (Fröhlich and Wanner 2008), without user interaction.

Hence, with all these changes made in the original version of the SIP protocol, there is a great economy of memory, achieving a final system image with a small size. Then, this proposed implementation can be used even in the smallest and simplest embedded systems present today.

Copy Rights

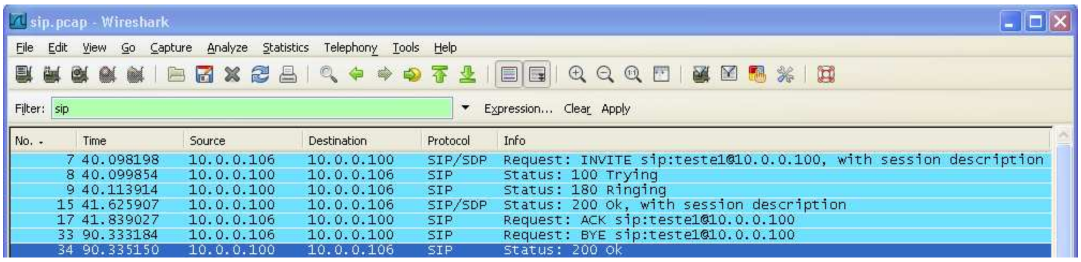
4. RESULTS

To evaluate the applicability of the proposed SIP implementation for embedded systems, we have measured the generated library. Table 1 shows the size of the version created, comparing to other well-known SIP implementations, like oSIP library (Moizard 2002) and the part referring to SIP in Asterisk (Digium 1999). All codes were compiled for the IA32 architecture, using the GNU gcc compiler version 4.0.2 and measured with the ia32-size tool, version 2.16.1. The results show that, with all the reductions made, it was obtained the final library size of 87,450 bytes, almost three times smaller than oSIP library and eight times smaller than Asterisk. The final image size, with EPOS and SIP version, reached 118,532 (116 Kbytes), much less than the others, even without considering the several megabytes of operating system overhead.

Table 1. Size of SIP implementations in bytes.

Section	SIP	libosip	Asterisk
.text	81,074	232,914	609,678
.data	6,360	3,220	7,912
.bss	16	1,172	67,972
TOTAL	87,450	237,306	685,562

In order to demonstrate the functionality, a communication test between two machines was performed. On one side, the EPOS operating system with the SIP implemented, running on VirtualBox virtual machine. On the other side, a machine with a SIP phone ready to make and receive calls. It was tested the startup and shutdown of sessions across the two machines. Figure 1 shows the test output in the format of messages flow through the graphical interface of the Wireshark program. The IP 10.0.0.106 represents the virtual machine with EPOS, while 10.0.0.100 a machine running Windows XP operating system and X-Lite SIP phone.


sip.pcap - Wireshark
File Edit View Go Capture Analyze Statistics Telephony Tools Help
Expression... Clear Apply

Filter: sip

No.	Time	Source	Destination	Protocol	Info
7	40.098198	10.0.0.106	10.0.0.100	SIP/SDP	Request: INVITE sip:teste1@10.0.0.100, with session description
8	40.099854	10.0.0.100	10.0.0.106	SIP	Status: 100 Trying
9	40.113914	10.0.0.100	10.0.0.106	SIP	Status: 180 Ringing
15	41.625907	10.0.0.100	10.0.0.106	SIP/SDP	Status: 200 Ok, with session description
17	41.839027	10.0.0.106	10.0.0.100	SIP	Request: ACK sip:teste1@10.0.0.100
33	90.333184	10.0.0.106	10.0.0.100	SIP	Request: BYE sip:teste1@10.0.0.100
34	90.335150	10.0.0.100	10.0.0.106	SIP	Status: 200 Ok

Figure 1. Initialize and finalize a session with SIP in Wireshark.

5. CONCLUSION

This paper proposed adaptations to the SIP protocol to make it feasible in resource-constrained embedded systems. By using only a subset of the total group of methods present in the standard, as well as a subset of header and request fields, it was possible to generate a small version of the SIP protocol without compromising its functionalities.

The results have shown that the proposed SIP miniature version consumes less than 120Kb of code and data. Through a test scenario, we also demonstrated that it is able to communicate to any SIP phone, starting and ending a session.

As future work, we intend to implement the developed version in a real embedded system and to study other necessary protocols for communication, such as RTP.

Copy Rights

REFERENCES

Cheng, B. et al, 2006. Context-Aware Gateway for Ubiquitous SIP-Based Services in Smart Homes. *International Conference on Hybrid Information Technology*, Vol. 2, November 2006, pp. 374-381.

Digium. 1999. *Asterisk – The Open Source Telephony Projects*. [Online] Available at: http://www.asterisk.org/ [Accessed 15 July 2010].

Fröhlich, A. A., 2001. Application-Oriented Operating Systems. 1st ed. GMD – Forschungszentrum Informationstechnik, Sankt Augustin, Germany. ISBN: 3-88457-400-0.

Fröhlich, A. A. and Wanner, L. F., 2008. Operating System Support for Wireless Sensor Networks. *Journal of Computer Science*, Vol. 4. No. 4, pp. 272-281.

Kwak, J, 2007. Ubiquitous Services System Based on SIP. *IEEE Transactions on Consumer Electronics*, Vol. 53, No. 3, August 2007, pp. 938-944.

Lakay, E. T.; Agbinya, J. I., 2005. SIP-based content development for wireless mobile devices. 1st International Conference on Computers, Communications, & Signal Processing with Special Track on Biomedical Engineering, November 2005, pp. 130 - 134.

Liao, J. et al, 2009. A demand-driven parsing method for SIP offload in home network. *IEEE Transactions on Consumer Electronics*, Vol. 55, No. 3, August 2009, pp. 1308-1314.

Moizard, A. 2002. *The GNU oSIP library*. [Online] (Updated 03 October 2008) Available at: http://www.gnu.org/software/osip/ [Accessed 13 July 2010].

Oh, Y. et al, 2006. Design of a SIP-based Real-time Visitor Communication and Door Control Architecture using a Home Gateway. *IEEE Transactions on Consumer Electronics*, Vol. 52, No. 4, November 2006, pp. 1256-1260.

Roach, A. B., 2002. Session Initiation Protocol (SIP)-Specific Event Notification. RFC 3265.

Rosenberg, J. et al, 2002. SIP: Session Initiation Protocol. RFC 3261.

A BLENDED LEARNING APPROACH IN SOFTWARE DEVELOPMENT COURSE: A CASE STUDY

Kechi Hirama
University of São Paulo, Escola Politécnica,
Department of Computer and Digital Systems Engineering,
Av. Prof. Luciano Gualberto, trav. 3, 158
05508-900 – São Paulo – SP – Brazil

ABSTRACT

Distance education is a reality today. Many educational organizations have adopted this approach, and for some of them, it has been the only format offered. Distance education is often compared with traditional education; however, due to the advantages and disadvantages of both approaches, it is very common to find the two formats complementing each other. Rather than simply comparing the two, another important method is the blended learning approach, which exploits the advantages and minimizes the disadvantages of each approach. This work presents an environment of distance education by way of a case study of a successful software development course that was offered to a large Brazilian financial organization. Course data were collected and analyzed. The outcome was generally satisfactory, but this study offers suggestions for some additional improvements.

KEYWORDS

Blended learning; collaborative learning; distributed learning environments.

Copy Rights

1. INTRODUCTION

Distance education has traditionally been used to give educational opportunities to students for whom the conventional educational models are not suitable, such as people at work, at home, living in isolated areas, or people with physical limitations [Watabe et al., 1995].

Many discussions about traditional and distance education models have been published. Each one has pros that can be maximized and cons that can be minimized by combining the two models into an approach known as Blended Learning. In this approach, the traditional (face-to-face) and distance models are combined, using the best characteristics of both, such as social support (traditional model) and flexibility (distance model) [Hentea et al., 2003], [Ranganathan, et al., 2007]. According to Hoic-Bozic et al. (2009), blended learning is based on various combinations of classical face-to-face lectures, learning over the Internet, and learning supported by other technologies, aimed at creating the most efficient learning environment. Itmazi and Tmeizeh (2008) discussed the benefits, issues and necessities by adopting blended learning in campus-based universities.

In recent research, student performance in distance education has increased. According to Means et al. (2009), the results are better in distance than in traditional education. Zhang et al. (2004) presented similar results, in which students in distance education performed significantly better than traditional ones when they faced two disciplines prepared for both models by the same instructors. Also, Mukti et al. (2005) found that students who participated in online collaborative learning course had performed significantly better result than students who studied in traditional way.

This work presents a learning environment based on the blended learning approach and the results of a successful case study in software development course in a large Brazilian organization in the financial area.

2. PARTICIPATION OF STUDENTS AND INSTRUCTORS

According to Davis (1993), students learn much more when they are involved in the education process. Participation should be viewed as a two way road, i.e., the students should be encouraged to communicate their opinions, read and react to colleagues' opinions as well. Also, distance education encourages students to participate in the education process by exploiting and using technological resources, and promotes interactions between students and instructors that do not happen in the traditional model [Pendergast, 2008]. Student participation is the vital component for any education model. In courses in which the focus is the teaching and learning of theories and concepts, as opposed to abilities and experiences, the key activity in which the students can participate is discussion [Clark, 2000].

In the distance education model, due to there not being face-to-face visual contact, participation would be stimulated using available Internet applications to generate collaboration among people. Nevertheless, it remains difficult to motivate students to participate in the discussion, especially without a specific class time where the student chooses his own schedules. Further, the instructor would also be working in his own schedule.

An important issue is to promote participation, but it should be taken into account that people are fundamentally different. According to recent research about the personality factor's effects on participation in distance education, some students performed better than others because they felt more comfortable elaborating questions in a computer discussion. On the other hand, personality traces like self-motivation affect learning both in distance and traditional education. Self-motivated students participated more than others in the class. However, shy students had much more time to interact with colleagues and instructor [Huang, 2009].

The instructor's role in the distance education environment is as important as the role of student differences. Although similar to that in traditional education, the instructor's role in distance education requires new capabilities and strategies. If the participation of the students and instructors is a very important factor of success in distance education, it is necessary to know how students should be evaluated.

The student learning evaluation is fundamental for verifying whether the didactic proposals were assimilated. The level of learning indication is essential in both traditional and distance courses. Although in traditional courses, due to face-to-face approach, there is immediate feedback from students according to stimulus-response formats that can be developed in the classroom, the challenges are different in distance courses. These differences can be overcome if the distance classroom is supported with updated technological resources like Web 2.0. Some examples of Web 2.0 applications are Google, Wikipedia, YouTube and Orkut.

Copy Rights

3. A DISTANCE EDUCATION ENVIRONMENT

Nowadays, a distance education environment applies several modern technological resources to support the pedagogical objectives of the institution. The infrastructure that supports such environments is called Computer Mediated Communication (CMC). CMC is a generic term that incorporates all forms of communications between individuals and among groups via networked computers [Naidu and Järvelä, 2006].

There are two forms of communications in distance education environment: synchronous and asynchronous. In the synchronous model all students and instructor are logged on the same time and communicate directly and virtually with each other. In the asynchronous model the communication between participants does not occur simultaneously [Itmazi and Tmeizeh, 2008].

Examples of environments are Angel Learning System of Angel Learning [Pendergast, 2008], LearnLink of iLinc Inc, [Koppelman and Vranken, 2008] and QUEST (Quest Environment for Self-managed Training) [Regueras et al., 2009] and aLF (active Learning Framework) of UNED [Pastor et al., 2009]. Figure 1 presents a possible physical configuration for a distance education environment. The Internet/Intranet is the most widely spread technology nowadays that allows many interaction and knowledge dissemination formats.

The educational environment is a website (portal) based on Web 2.0 technology.

4. THE DIDACTIC PROPOSAL

The didactic proposal is based on the collaborative learning method, through which students interchange opinions and discuss in groups, and acquire, develop and amplify their knowledge based on the course subject and specific themes. Also, the asynchronous model was defined for distance education environment discussed in section 3.

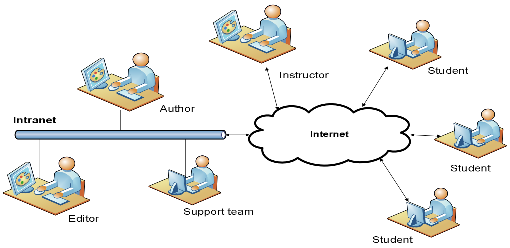


Figure 1. A distance education environment.

The collaborative learning method is applied on a specific theme in the Information Technology area. The theme is organized into chapters. The student should read the chapters and develop commentaries on or clear doubts about the read chapter. Each commentary is evaluated by the instructor to consider whether or not it is relevant to other students. The collaborative format allows the discussions to create a common sense about the chapter. The information interchange among students allows each one to perceive the adherence level of his understanding in relation to the others through their evaluations of participations. The individual evaluation is composed of instructor evaluation and the group of students' evaluations of the posted commentary.

At the end of the work week in a set of chapters, the instructor closes the discussions and reports the important aspects discussed by students.

Some collaborative mechanisms are available in the learning environment. Beyond commentaries and evaluations about posted commentaries in a blog format, there is the possibility of clearing doubts in a specific section named Questions and Discussions. Students can post questions in this section, creating a channel in which colleagues can post their answers. The subjacent idea is that by trying to answer the question, the student consolidates the knowledge. The instructor does not answer the questions directly, but he stimulates the participation of students. A student can post several questions.

Another form of collaboration is through challenges posted by the instructor in a section named Instructor's Challenges. A challenge typically represents an aspect that was not adequately explored by students and deserves some deepening or knowledge amplification. Through regular monitoring of the discussions and commentaries of the students, the instructor can detect gaps in information that should be filled. This becomes the raw material for elaborating a challenge. The challenge can result from a general view that stimulates the students to a deeper reflection based on themes discussed in the course and their day-to-day realities.

The other form is established through a chat named Group Conversation. Here, the instructor establishes a date and a period to clear students' doubts about general aspects of the course.

Finally, the environment supports the e-mail systems that allow communications among students, instructor and editors. The instructor also uses this system to keep in touch with the students.

Figure 2 presents the portal of this didactic proposal.

5. A CASE STUDY AND PERFORMANCE ANALYSIS

5.1 Case Study

A case study was performed in a software development course with 25 groups of a large Brazilian organization in the financial area. Each group was constituted by 18 students (managers, system analysts, data base analysts, system architects and programmers), 1 instructor and 1 editor. Therefore, the total was 450 participants from May to December 2009.

During this period, three presentations took place (Opening, Intermediate, and Concluding) with students, instructors, editors and the course coordinator. The opening meeting aimed to welcome the students and make them aware of the course objectives, didactic approach, portal structure and general recommendations. The intermediate meeting aimed to hear the students' opinions about their performance until that moment, to emphasize the best moments of the students´ participations and to present proposals for the course's final projects. The concluding meeting again aimed to hear the students´ opinions about their performance in the course, to emphasize the best moments of the students' participations and the groups' performance, and to present the abstracts of the final projects.

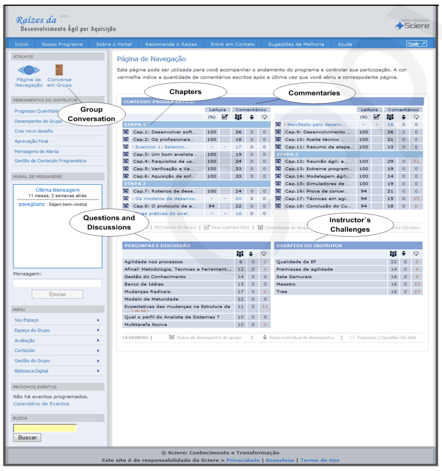
The course aimed at teaching a software development process to be implemented in the Information Technology area. The process is based on service outsourcing of software development, to which agile concepts and techniques are applied to guarantee the committed time, cost and product quality. The course was organized in chapters divided into three phases. The first was concerned with role definitions, centered in systems analysts. The second was related to the software development process and the third presented the agile concepts and techniques that can be applied at several moments during the product development.

5.2 Course Evaluations

The course had two evaluation types: individual and group performance and the course itself.

The individual and group performance was measured based on student participation in chapters reading, commentaries, questions, answers and opinions posted during the course. The minimum grade to be approved was 7.0. Each posted commentary by a student was evaluated by the instructor and by his colleagues. A commentary, question, answer or opinion was initially evaluated by the instructor as relevant to be discussed among the students. Next, each student could also evaluate the posted commentary on the following three aspects: was it helpful to understand some aspect, did it deepen their understanding and did it expand the discussion of theme.

Figure 3 presents the average student evaluations per group (Gi). It is verified that all groups reached the minimum average (7.0). The total number of students was 448 in 25 groups, 420 students (93.8%) were approved, 2 failed (0.4%) and 26 withdrew (5.8%).



Página de Navegação

Esta página pode ser utilizada para você acompanhar o andamento do programa e controlar sua participação. A cor vermelha indica a quantidade de comentários escritos após a última vez que você abriu a correspodente página.

CONTEÚDO PROGRAMÁTICO

	Leitura		Comentários		
	(%)	☑			
ETAPA 1					
Cap.1: Desenvolver soft...	100		26	0	0
Cap.2: Os profissionais...	100		18	0	0
Exercício 1: Selecion...	-	-	17	0	0
Cap.3: Um bom analista ...	100		19	0	0
Cap.4: Requisitos de us...	100		24	0	0
Cap.5: Verificação e Va...	100		33	0	0
Cap.6: Aquisição de sof...	100		33	0	0
ETAPA 2					
Cap.7: Roteiros de dese...	100		24	0	0
Os modelos de desenvo...	-	-	20	0	0
Cap.8: O protocolo de a...	94		22	0	0
...as práticas do anal...	-	-	16	0	0

	Leitura		Comentários		
	(%)	☑			
Manifesto pelo desenv...	-	-	18	0	0
Cap.9: Desenvolvimento ...	100		26	0	0
Cap.10: Aceite técnico ...	100		21	0	0
Cap.11: Resumo da etapa...	100		13	0	0
ETAPA 3					
Cap.12: Reunião ágil: a...	100		29	0	12
Cap.13: Extreme program...	100		19	0	0
Cap.14: Modelagem ágil:...	100		14	0	0
Cap.15: Simuladores de ...	100		19	0	0
Cap.16: Prova de concei...	94		21	0	0
Cap.17: Técnicas em agi...	94		15	0	15
Cap.18: Conclusão do Cu...	94		18	0	8

... | (%) Leitura do Grupo | Seus capítulos lidos | Comentários do Grupo ... ...ários não lidos

PERGUNTAS E DISCUSSÃO			
Agilidade nos processos	6	0	1
Afinal! Metodologia, Tecnicas e Ferrament...	12	0	2
Gestão do Conhecimento	14	0	0
Banco de Idéias	13	0	0
Mudanças Radicais.	17	0	6
Modelo de Maturidade	22	0	0
Expectativas das mudanças na Estrutura da ... NOVO	11	0	11
Qual o perfil do Analista de Sistemas ?	10	0	0
Multitarefa Nociva	10	0	3

DESAFIOS DO INSTRUTOR			
Qualidade da EF	22	0	2
Premissas de agilidade	14	0	6
Sete Samurais	16	0	9
Maestro	16	0	13
Tree	16	0	13

LEGENDAS | Status de desempenho do grupo | Status individual de desempenho | Respostas / Opiniões não lidas

Última Mensagem
11 meses, 2 semanas atrás
pavegliano : Sejam bem-vindos

Mensagem:

PRÓXIMOS EVENTOS
Não há eventos programados.
Calendário de Eventos

© Sciere: Conhecimento e Transformação
Este site é de responsabilidade da Sciere > Privacidade | Ressalvas | Termo de Uso

Figure 2. Portal of the didactic proposal.

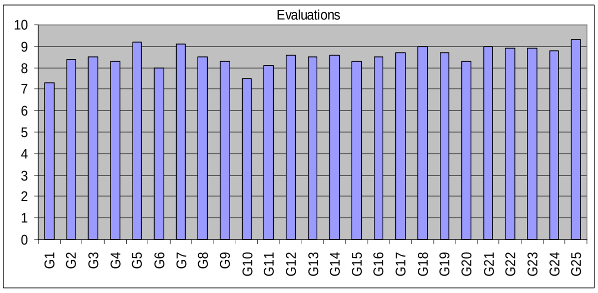


Figure 3. Average student evaluations per group.

The course evaluation took place at the concluding meeting, when a questionnaire was distributed to the students. The answered questionnaires were 253 (56.5%) out of 448 students that took the course.

The evaluation levels are 1 – Nothing (absentee, very unsatisfied), 2 – Low (insufficient), 3 – Medium (reasonable), 4 – Good (relevant) and 5 – Excellent (high, very satisfied). The questions were grouped into the following:

a) Learning Evaluation: the student's level of comfort with the learning process, their level of effort to follow the course pace, the relevance of the discussed themes, of learning, of professional life impact and of benefits to the organization. The average Learning Evaluation was 3.75 (almost good), where the maximum value was 3.99 (good) in Relevance of Theme and the minimum was 3.58 (between medium and good) in Necessary Effort. The maximum value is justified, because the software development process (the course's theme) will be implemented in the organization, being therefore very relevant to the student. In relation to the minimum value, it was expected, because the estimated dedication to the course per student was 0.5 hour per chapter. Being that three new chapters were available weekly, each student could read and post one commentary about the text read in 1.5 hour. Still, due to the discussions and interactions with colleagues, the reported average by students was 4.92 hours weekly or 0.7 hour per day of dedication to the course.

b) Study in Groups Method Evaluation: the group contribution level to the student learning, the student contribution level to the group, the learning level with the texts provided, the importance level of further reading material and site links, the importance level of presentations and the importance level of evaluations to students' learning. The average Study in Groups Method Evaluation was 3.59 (between medium and good). The maximum value was 3.93 (good) in Your Learning through Chapter Texts, showing that the texts provided useful and relevant information. The minimum value 3.33 (medium) in Presential Meetings showed that the discussed topics in the meetings did not contribute or were not relevant to student learning. This is an item that needs to be analyzed to make meetings more attractive. Another factor was the low attendance in meetings, about 40% per study group.

c) Management and Orientation of Study in Groups Evaluation: the importance level of the instructor's orientations during the discussions, support level of the instructor to the group's searching for its own answers, the importance level of e-mails with weekly reports to motivate participation, the importance level of the closing commentaries in each chapter, the service satisfaction level of the portal operational management and the agility level of the support team. The average Management and Orientation of Study in Groups Evaluation was 3.96 (good); the maximum of 4.09 (good) in Instructor Allows the Group Members to Search their Own Answers showed that the instructor was careful to support, without excessively interfering, in answering the posted questions. The minimum 3.83 (almost good) in Instructor's Orientation in Discussions indicated that the instructor participation could be more active in clearing the doubts or organizing the discussions. According to the philosophy of the course, the interference level of the instructor is difficult to define, since, on the one hand, the instructor should be faithful to the theme and to the current chapter and, on the other hand, unexpected situations emerge that require the instructor's immediate response, e.g., when the subject of discussion was an organizational issue.

d) General Evaluation of Portal: the student's satisfaction level with the course, motivation level to continue the course and to continue updating their knowledge, to study other themes and the intention level to recommend the course to friends. The average General Evaluation of Portal was 3.96 (good) and the maximum 4.05 (good) in Are You Satisfied with the Program? Was it Worthwhile? and Would You Recommend the Course to Friends? showed that the course objectives were reached and therefore they would recommend the course to their friends. The minimum was 3.86 (almost good) in Was the Course Pleasant? Did You Feel Stimulated to Continue? showing that the students considered the course adequate and its format allowed them to feel motivated to participate and to continue their studies.

6. CONCLUSION

This work presents a case study of a course successfully applying the distance education model. Widely discussed in theory, distance education is also a practical reality today. Traditional and distance education will continue to meet different sets of needs, but they can perform important complementary roles. The blended learning approach would be the best combination, benefiting from the advantages and minimizing the disadvantages of both models.

In this work, the collaborative distance education approach predominates, but the presentations with students, at the start, in the middle and at the closing of the course, were important for analyzing the general feeling toward the course. The reported case study corroborates the potential of distance education. The data collected and analyses performed inspire future work in the creation of adaptable environments allowing the personalization of a terminal that explores the preferences and history of students.

According to Dagger et al. (2003), personalization of learning is potentially beneficial in terms of time, money and effectiveness. Hamad et al. (2008) discussed a framework of the system that implements learning styles that can guide students to the study techniques that are most likely to be effective to them.

ACKNOWLEDGEMENT

I would like to thank the professionals of Sciere Consultores Associados Ltda., especially my colleague Edson Fregni, Ph. D. (President), for giving me the opportunity to participate in his team, Priscilla B. Avegliano (editor), Denise L. Gastaldo, Christiane M. Baptista, Marcel Smetana, Marta Bisca and Henrique S. Fugita (instructors) and support team for their effort and dedication that allows providing the courses.

REFERENCES

Clark, M., 2000.Getting Participation Through Discussion. *Proceedings of the Thirty-First SIGCSE Technical Symposium on Computer Science Education*. pp. 129-133.

Davis, B. G., 1993 *Tools for Teaching*. Jossey-Bass Publishers. San Francisco, 1993.

Dagger, D. et al., 2003. Towards “anytime, anywhere” Learning: The Role and Realization of Dynamic Terminal Personalization in Adaptative eLearning. *World Conference on Educational Multimedia, Hypermedia and Telecommunications*. Ed-Media, 2003.

Hamad, A. A. et al., 2008. Integrating ´Learning Style´ Information into Personalized e-Learning System. *In IEEE Multidisciplinary Engineering Education Magazine*, Vol. 3, No. 1, pp. 2-6.

Hentea, M. et al., 2003. A Perspective on Fulfilling the Expectations of Distance Education. *Proceedings of the 4th Conference on Information Technology Curriculum*. pp. 160-167.

Hoic-Bozic, N. et al., 2009. A Blended Learning Approach to Course Design and Implementation. *In IEEE Transations on Education*, Vol. 52, No. 1, pp. 19-30.

Huang, I., 2009. The Effects of Personality Factors on Participation in Online Learning. *Proceedings of the 3rd International Conference on Ubiquitous Information Management and Communication.* Korea. pp. 150-156.

Itmazi, J. A. and Tmeizeh, M J. 2008. Blended eLearning Approach for Traditional Palestinian Universities. *In IEEE Multidiciplinary Engineering Education Magazine*, Vol. 3, No. 4, pp. 156-162.

Koppelman, H.; Vranken, H., 2008. Experiences with a Synchronous Virtual Classroom in Distance Education. *Proceedings of the 13th Annual Conference on Innovation and Technology in Computer Science Education*. Spain. pp. 194-198.

Means, B. et al., 2009. *Evaluation of Evidence-Based Practices in Online Learning: A Meta-Analysis and Review of Online Learning Studies*. U.S. Department of Education. USA. 2009.

Mukti, N. A. et al, 2005 Hybrid Learning and Online Collaborative Enhance Students Performance. *Proceedings of the Fifth IEEE International Conference on Advanced Learning Technologies (ICALT´05).* Kaohsiung, Taiwan, pp. 481-483.

Naidu, S. and Järvelä, S., 2006. Analyzing CMC content for what? *In Computer & Education*, 46, pp. 96-103.

Pastor, R. et al., 2009. Virtual Communities Adapted to the EHEA in an Enterprise Distance e-Learning Based Environment. *In Lecture Notes on Computer Science – LNCS*. Vol. 5621. pp. 488-497.

Pendergast, M. O., 2008. Three Short Cases for Use in Online Introduction to Computer Information Systems Courses. *In ACM SIGITE Newsletter*. Vol. 5, No. 1, pp. 6-15.

Rada, R. and Hu, K., 2002. Patterns in Student-Student Commenting. *In IEEE Transactions on Education*. Vol. 45, No. 3. pp. 262-267.

Ranganathan, S. et al., 2007. Hybrid Learning: Balancing Face-to-Face and Online Class Sessions. *Proceedings of the 2007 Southern Association for Information Systems Conference*. pp. 178-182.

Regueras, L. M. et al., 2009. Effects of Competitive E-Learning Tools on Higher Education Students: A Case Study. *In Transations on Education*, Vol. 52, No. 2, pp. 279-285.

Watabe, K. et al., 1995. An Internet Based Collaborative Distance Learning System: CODILESS. *In Computers Education*, Vol. 24, No. 3, pp. 141-155.

Zhang, D. et al., 2004. Can E-Learning Replace Classroom Learning? *In Communications of the ACM.* Vol.. 47, No. 5. pp. 75-79.

Copy Rights

LANGUAGE MODELING IN THE CONTEXT OF A MDE PROCESS

Thanh Thanh Le Thi and Pierre Bazex
IRIT – Université Paul Sabatier - 118, route de Narbonne - 31062, Toulouse Cedex 9

ABSTRACT

Generally, the components of critical systems have strict requirements since any mistake during operation may harm all related systems. In this context, it is important to guarantee that these components have required technical qualities and to insure that the intention of analyst in the system modeling step is well respected in the code generation phase.

Therefore, we focused on modeling the language's syntax and its operational and axiomatic properties in UML/OCL to provide analysts/designers a clear and unambiguous definition of the target language. Moreover, we define an incremental process for the code generation with the well-defined target language as an intermediary step. This method provides analysts/designers a solution to verify and to prove at each step of the code generation phase that the application requirements have been well interpreted and taken into account.

This research has been applied in the fields of aeronautics and space in the DOMINO ANR project (2007-2009).

KEYWORDS

Syntax, Semantics, Language modeling, UML/OCL, Incremental process, Code generation

Copy Rights

1. GRAMMAR TECHNOLOGIES IN THE CONTEXT OF MODELWARE

Model Driven Engineering (MDE) puts models as the centre of the whole process, which begins from translating the requirements of the application into conception models and ends with the code generation from these designed models. The requirement modeling activity is considered as the most important phase in the process whereas the code generation is reduced as a simple step that is directly done from the designed models. However, this code generation step is one of the most critical ones. Actually, it is the transition between the modeling, which represents requirements with the abstract artifacts and a set of programs, which represents all technical and concrete artifacts of a language with its own precise syntax and semantic.

In order to provide analysts/designers an opened environment with the appropriate controls and verification to reduce the gap between these two fields, we propose the use of the languages' syntax and semantics' modeling as an intermediary step in the code generation phase. This intermediary step is the component modeling in UML/OCL regarding the target language's properties. It provides the analysts/programmers all possibilities to check if the component has the required qualities before the actual code generation. In particular, we also guarantee that the components verify all target language, platform's properties and all properties that we can specify at this level of modeling. Moreover, the analysts/designers can execute all or part of the designed components' models by integrating operational semantic of the target language. They can also statically verify the correctness of the designed models thanks to the axiomatic semantic.

To integrate this modeling level in the code generation phase, we firstly resume the compiler techniques in combination with the work in language modeling domain. There are many works in language modeling field: [Malenfant] aimed at precisely describing a language with its syntax and semantic in UML/OCL. [Alanen] discussed the bridge from modelware to grammarware, in the context of mapping MOF meta-models to context-free Grammar whereas [Wimmer] have presented the bridge from grammarware to modelware. [Kelsen] used Alloy based method to formally model a language. The two tools Sintak [Muller] and TCS [Jouault] make the correspondence between the abstract and concrete syntax of a language. XText (http://www.eclipse.org/Xtext/) is another tool which gives us the possibility to easily create a language parser, based on its grammar. In our study, we inspire the work in [Malenfant] to model a language in UML/OCL with its static, operational and axiomatic properties and then use it in the development process.

The paper will be presented as the following: sections 2 and 3 describe the language modeling which focus respectively on the static, operational and axiomatic semantic modeling. Section 4 presents our exploitation of the language modeling in the definition of a model transformation pattern and of the incremental code generation. We conclude the article with some ideas on the upcoming works.

2. LANGUAGE MODELING – CODE MODELS EXECUTION

2.1 Example of a Language Grammar – the L Language

We present here a simple language which is used to illustrate the language modeling. Traditionally, a language is described by a grammar which constitutes of a group of symbols and syntax rules (SR). Each rule describes the syntax construction of a particular symbol at the left hand side, called a non-terminal symbol. Each predefined type of the language is presented as a terminal symbol which has no syntax construction's definition. There are two different rule types: the AND rule with the "," between two symbols, the OR rule with the "|" between two symbols. The following table shows the syntax constructs of the L language:

-- SR for the declarative and instruction parts	-- SR for the instructions	True →
Prog_L → DP , IP	*Inst* → Skip | Seq | Affect |Test | Loop	False →
DP → VarDec*	Skip →	C_Nat →S
IP → Inst	Seq → Inst, next: Inst	Null → Exp
-- SR for the variables	Affect → Variable, Exp	*ExpBin* →Add | Sub | Mult | Div | Xor | Inf
DecVar → Variable, TP_L	Test → Exp, then : Inst, else : Inst?	Add → exp1: Exp, exp2 : Exp
Variable → S	Loop → Exp, Inst	Sub → exp1: Exp, exp2 : Exp
-- SR for the predefined types	-- SR for the expressions	Mult → exp1: Exp, exp2 : Exp
TP_L → Bool | Nat	*Exp* → Variable | Const | Null | ExpBin	Div → exp1: Exp, exp2 : Exp
Bool →	*Const* → C_Bool | C_Nat	Xor → exp1: Exp, exp2 : Exp
Nat →	*C_Bool* →True | False	Inf → exp1: Exp, exp2 : Exp

This grammar is complemented with a set of static, operational and axiomatic properties. The static semantic defines the well-typing relations for all symbols of the language. The operational semantic of a language describes the dynamic behaviors of its constructs as a transformation of memory's state. The axiomatic semantic defines the interpretation of program's instructions as the transformation of assertions which specify the memory's properties. We do not mention here the description of the corresponding semantic for the L language, please refer to [Paulin-Mohring] for more information.

2.2 Code Models Execution

To model a language, we start with the syntax modeling – presenting all syntax constructs by the meta-classes. The relations between the syntax constructs are presented by the composition link (for AND rules) and by the heritage link (for OR rules), as described in our previous work [Le Thi]. We obtain then a meta-model for the syntax part of a language.

Concerning the modeling of a language's semantics, we use OCL and an action language. The static semantic determines the type of an expression and calculates the well-typing for the instructions. We have modeled the static semantic with the corresponding functions: *Exp::type() : TP_L* returns the type of an expression as an object whereas *Inst::ok():Boolean* returns a boolean value.

In case of the operational semantic, we need to model the runtime environment where the relations between the predefined types of the language and that of the environment are established. We model this environment as a set of memories which couple a *Variable* and a semantic value *SemVal*. In this modeling, *SemVal* holds the semantic domain of the *Variable* and its type in L language. The *Bool* and *Nat* type of the L language will be respectively mapped to *V_Boolean* and *V_Entier*. The operational semantic is then modelled with the evaluation and execution functions of different expressions and instructions. The *Exp::eval(env: EnvExec):SemVal* evaluates the expression and returns a semantic value whereas the *Inst::exec(env: EnvExec):EnvExec* transforms the environment into a new state. In the following table, we list the modeling of the static and operational semantic of some syntax constructs:

Copy Rights

```
-- Static semantic modeling in OCL
Exp::type(): TP_L
Add ::type() :  TP_L = if  self.exp1.type().oclIsTypeOf( Nat ) and
                           self.exp2.type().oclIsTypeOf( Nat )
                       then new_Nat()   else oclUndefined( Nat )
                       endif
…
Inst:: ok() : Boolean
Seq ::ok() :   Boolean = self.inst.ok() and self.next.ok()
Affect::ok() : Boolean = (self.variable.type().oclIsTypeOf( Nat ) and
                            self.exp.type().oclIsTypeOf( Nat  ) ) or
                        (self.variable.type().oclIsTypeOf( Bool ) and
                            self.exp.type().oclIsTypeOf(Bool ) ) …
```

```
-- Operational semantic modeling in OCL/action language (USE language)
Exp ::eval( env : EnvExec ) :   SemVal
Add ::eval( env : EnvExec ) : SemVal =
                          self.exp1.eval( env ).oclAsType( V_Entier ).add(
                          self.exp2.eval( env ).oclAsType( V_Entier )
)
…
Inst::exec( env: EnvExec): EnvExec
Seq ::exec( env : EnvExec ) :EnvExec = self.next.exec( self.inst.exec( env ) )
Affect ::exec( env : EnvExec ) :EnvExec =
                     env.majVar( self.variable.copy(),  self.exp.eval( env ) )
…
```

3. LANGUAGE MODELING – CODE MODELS VERIFICATION

Axiomatic semantic is used to prove the correctness of computer programs by static analysis. It defines the interpretation of program's instructions as the transformation of assertions which specify the memory's properties. These assertions can be coupled as Hoare triple {P}*Inst*{Q} that is interpreted as: the execution of the *Inst* instruction in an initial memory which satisfies the assertion P conducts to a memory satisfying Q. We can call P as pre-conditions and Q as post-conditions. However, it is difficult to prove Q from the execution of *Inst* in the initial memory satisfying P. Therefore, we prefer to calculate the weakest precondition (WP) P' from Q and *Inst*. We have {P'}*Inst*{Q} as a valid Hoare triple and need to prove $P \Rightarrow P'$ to verify {P}*Inst*{Q}.

Using the same principles, we propose to model the axiomatic semantic by implementing the WP function in OCL/action language for each kind of the language's instruction. About the second step of proving $P \Rightarrow WP$, we can refer to a specific environment, such as the formal B environment. This allowed us to concentrate our work in the field of MDE. We focus on showing how all these operational and axiomatic properties of languages can be implemented directly in the UML/OCL/action language. The idea is to maintain the homogeneity of the UML/OCL formalism throughout the whole development process. The only external call that analysts/designers need to do is to call a prover for the assertions of the $P \Rightarrow WP$ type, knowing that these assertions contain only the symbols from the grammar of the language.

We model an assertion as an expression Exp. To be able to describe these assertions, we have extended the Boolean expression by adding three operators – *Imp* for "imply"(⇒), *Eg* for "equality" and *And* for the "and" logic. The WP is modeled as the WP function *Inst::WP(q:Exp):Exp* for each kind of instructions:

```
 Inst::WP(q: Exp) : Exp
Seq ::WP( q : Exp ) :       Exp =       inst.WP( next.pfp( q.copy()  ) )
Affect ::WP( q : Exp ) :    Exp =       q.copy().substitution( self )  …
```

Take as example the following code simplification: 5 * 3 + 5 * 4 = 5 * (3 + 4) which is generally noted as: *((exp11 * exp12) + (exp21 * exp22)) And (exp11 ≡ exp21) = exp11 * (exp12 + exp22)*. This typical code simplification is simple. However, we need to guarantee that these two source/target expressions give the same operational semantic. In this case, we use the axiomatic approach to prove the correctness of the transformation. In this example, if we call the source expression as x and the target expression as y, we need to prove the following Hoare triple:

P := { x.oclIsTypeOf(Add) ∧ x.oclAsType(Add).exp1.type().oclIsTypeOf(Mult) ∧ x.oclAsType(Add).exp2.type().
oclIsTypeOf(Mult) ∧ x.exp1.exp1.ident(x.exp2.exp1) }

y := new_Mult(x.exp1.exp2.copy(), new_Add(x.exp1.exp2.copy(), x.exp2.exp2.copy())

Q := { y.oclIsTypeOf(Mult) ∧ y.oclAsType(Mult).exp2.type().oclIsTypeOf(Add) ∧ x.eval(env) = y.eval(env)

We present the calculation of the weakest pre-condition for this example in a simplified way for better understanding of the approach. The following sequence of assignments presents the codes for this transformation:

```
(aff1) a := x11;   (aff2) b := x12;   (aff3) c := x22 ;   (aff4) y := a * ( b + c );
```

The WP calculation for these four assignments (aff1)→(aff4) can be done as the following :

Copy Rights

WP([(aff1), (aff2), (aff3), (aff4)] Q) = WP([(aff1), (aff2), (aff3)] WP ([(aff4)] Q)) =
WP([(aff1), (aff2), (affs3)] (eval(x) = eval(a * (b + c))), by replacing y with a * (b + c)

With the same manner of WP calculation for the three resting assignment, we obtain the corresponding proof obligation: $x11 = x21 \Rightarrow eval(x) = eval(x11 * (x12 + x22))$. In this proof obligation, the constraint is expressed on the source variables (x11, x21, x22) and not on the target ones (a,b,c). We have developed in B formalism this proof obligation where the assertion have well translated what we have in the OCL formalism - structural constraints on the variables (1) and the corresponding calculated proof obligation (2):

```
…
ASSERTIONS
!(env, eval, x11, x12, x21, x22, add1, mm1, mm2,mm3, add2 ).(
//Structural constraints on the variables (1)
 x11: Exp &…& add1: Add & add2 : Add & mm1 : Mult & mm2 : Mult & mm3 : Mult &{(add1 |-> mm1),(mm1 |->x11), (mm2 |->x21), (mm3 |->x11),
(add2 |-> x12)} <: exp1& {(add1 |-> mm2),(mm1 |->x12), (mm2 |->x22), (mm3 |->add2), (add2 |-> x22)} <: exp2  & env: VARI-->NAT & isEval(eval)
//The calculated P=>WP(2)
=> (( eval(env, x11) = eval(env,x21) )=>( eval(env,mm3) = eval(env, add1)) ))
ABSTRACT_VARIABLES Exp, OB , Add , Mult, ConstE, exp1 , exp2 , ctNat, VARI
DEFINITIONS
//express the structure  of the meta-model fragment, concerning Exp
inv == (Exp <: Expression & OB <: Exp & … & exp1 : OB --> Exp & exp2 : OB --> Exp & ctNat : ConstE --> NAT );
//express the operational semantic of different kinds of Exp
isEval(eval) == ( eval: (VARI --> NAT) * Exp --> NAT & !(env,add).(env:VARI -->NAT & add:Add => eval(env,add) = eval(env, exp1(add)) +
eval(env, exp2(add)) ) &...& !(env,cte).( env:VARI --> NAT & cte:ConstE => eval(env,cte) = ctNat(cte)) )
INVARIANT inv …
```

In this example of the factorization, we need to transform in the B formalism not only the corresponding meta-model fragment (e.g. *Exp* and its specializations) but also the definition of the operational semantic (the evaluation relations). Therefore, we can describe the assertions on the same operational semantic of the source and target model.

4. APPLICATION IN MDE PROCESS

4.1 Model Transformation Pattern

A model transformation is consisted of “elementary” ones. To prove the correctness of the whole model transformation, we propose to formalize each elementary transformation as a transformation pattern and to prove it in using the axiomatic semantic and a prover as previously described in section 3. We have inspired the work of [Nemo] to formalize each elementary model transformation as a pattern <EMM_s, PC_s, T, EMM_t, PC_t> in which the abbreviations are described as the following:

The effective source/target meta-models (EMM_s & EMM_t): an effective meta-model is a fragment of the meta-model which is consisted of the corresponding meta-class for the transformation. In the above factorization example, EMM_s and EMM_t are the L language meta-model fragment concerning the *Exp* and its specializations. As the EMM_s and EMM_t are fragments of the source/target meta-models, we need to verify that EMM_s and EMM_t are included in the corresponding meta-model [Nemo]. Otherwise, we can use the meta-model pruning approach proposed by [Sen] to guarantee this inclusion relation.

The constraints on EMM_s and EMM_t – PC_s and PC_t, called as conformity predicates in [Nemo]. Generally, these are the properties on the meta-model that cannot be structurally expressed.

The transformation T.

If we want to prove the correctness of the pattern, we need to take into account not only the structure of the source/target but also their operational and axiomatic semantic. For each pattern, we translate the structure of the pattern into B along with the operational and axiomatic semantic. And then, we can prove the correctness of the pattern with the help of the B prover.

4.2 Incremental Code Generation Process

Figure 1 describes the general architecture of the development process. We have detailed the code generation with another intermediary step to have a so-called “executable” model before going to the model of the target language. We describe shortly in the sub-sections each intermediary step.

Copy Rights

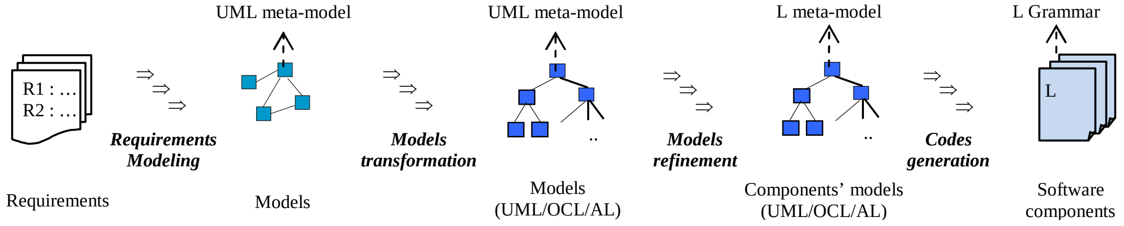


Figure 1. Architecture of an application of the incremental code generation process

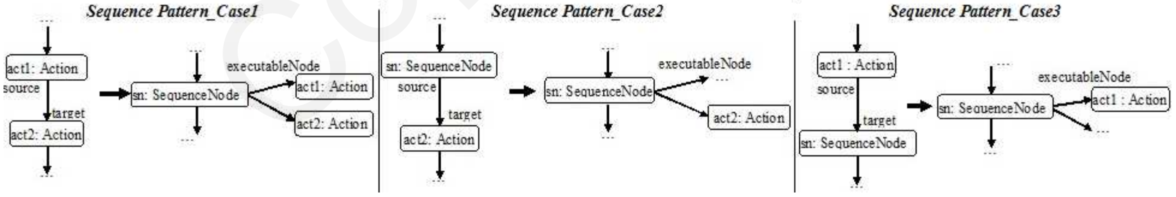
4.2.1 First Intermediary Step – towards an "Executable" Model

In this step, we focus on the "executable" criterion of the designed models which means that the behavioral part of the designed model can be translated into an executable program. With all the description possibilities given by the activity diagram, we can easily model a diagram that can't be translated into a programming language. Actual controls in almost tools, however, cannot detect such possible "bad-design". For this reason, we aim at translating the behavioral part of designed model into a hierarchical model where we can guarantee the criterion of its execution in a programming language. In our study, we focus on the algorithmic part of the activity diagram, without taking into account the synchronization and parallelism parts. In this case, the activity diagram is defined as a set of nodes and links between them where we are interested in the sequencing of the actions and in the use of Decision-MergeNode to express the branches. We define here the meta-model of hierarchical activity diagram (HAD) which corresponds to the following rules:

Table 1. Extraction of the syntax rules for the hierarchical activity diagram

HAD	→ ExecutableNode	StructuredActivityNode	→SequenceNode | ConditionalNode | LoopNode
ExecutableNode	→OpaqueAction | StructuredActivityNode	SequenceNode	→ExecutableNode+
OpaqueAction	→ …	…	

As seen in the table 1, we restraint the hierarchical activity diagram as an ExecutableNode which can be a specific action or a structured activity node – a sequence, conditional or loop node. So, we have eliminated the control flow concept which links the activity nodes. In fact, we use the structured activity nodes to replace the sequencing and branches that can be expressed with the control flows and the conditions on them. We have defined three model transformation rules that are formalized as the transformation patterns. We resume here the main point of first rule without presenting them as mentioned in the 4.1 due to the limitation of the paper. The Sequence pattern: For two consecutive actions linked by a control flow (with the *guard = true* and *weight = 1*), we transform this fragment into a sequence node which contains these two actions. In this pattern, we have the same operational semantic of the source/target models thanks to the preservation of the execution order of the actions as we can see in the following example cases.



The two other patterns concern DecisionNode-MergeNode fragment that we transform in a conditional node or in a loop node. We want to remark here that in this intermediary step, we aim at transforming the activity diagram into a hierarchical one. The operational semantic of these two are preserved as the order of the actions' execution is unchanged.

4.2.2 Second Intermediary Step – Components as a Target Language Model

This step translates the hierarchical models issued from the precedent step into the models of the target language. In this step, we control the validity of all target language's specific properties. Furthermore, with the help of operational semantic of the language, we can execute one part or the whole model in order to have a concrete vision of the corresponding final code execution. The axiomatic semantic, in the other hand, helps us to statically validate the program's model regarding the pre/post-conditions.

Copyrights

4.2.3 Injection of Concrete Syntax

Issued from the precedent step, we have the program model. This model conforms to the target language and has all required qualities. To obtain the final codes, we just need to inject into this model the concrete syntax of the final language. This operation is directly implemented at the language meta-model as a *toString()* operation or as a separated printer function. The injection of the concrete syntax from the abstract syntax is obviously much simpler and more precise than the generation of codes from the formalism at the design phase.

5. TOWARDS THE COMBINATION OF GRAMMAR/MODELS/PROOFS TECHNOLOGIES

Integrating the modeling of languages' syntax and semantic in UML/OCL into a model driven development process helps us to enforce the continuity between models and codes within the whole development process. Therefore, we have an extra control level where we can execute the models and check if the modeled components have all required qualities. With the axiomatic semantic modeling, we can statically verify the correctness of the program's model thanks to the external call to a formal environment of proofs (e.g. B tools). We have proposed in this article a methodology of using the three technologies – grammar, model, proofs. These three technologies form a solid technical base so that the incremental code generation process can guarantee that the application's requirements are well interpreted and taken into account from the model design step. In the context of DOMINO project, we have developed this work in the USE platform and have transformed it into Kermeta platform.

In this paper, we have mixed between the operations of the language with that of the assertions. In fact, we need to separate these two kinds of language in order not to "pollute" the target language meta-model. This is one of our actual objectives. Another one lies with the conversion from OCL to B. We aim at automatically doing this process in resuming the existing work of OCL to B conversion [Marcano].

REFERENCES

Alanen, M. and Porres, I. 2003. A Relation Between Context-Free Grammars and Meta Object Facility Metamodels. *Turku Centre for Computer Science TUCS Technical Report No,* vol. 606.

Jouault, F. , Bézivin, J. and Kurtev, I. 2006. TCS: a DSL for the specification of textual concrete syntaxes in model engineering. In *Proceedings of GPCE '06*, Portland, Oregon, USA, pp. 249-254.

Kelsen, P. And Ma, Q., 2008. A Lightweight Approach for Defining the Formal Semantics of a Modeling Language. In *Model Driven Engineering Languages and Systems (MODELS) 2008*, Toulouse, France, pp. 690-704.

Le Thi, T.-T. Modeling of programming languages in UML/OCL and application in a MDE process. In *IADIS International Conference Applied Computing,* Vol. I, p. 234-242, Rome, Italy, november 2009

Le Thi, T.-T.. L'Activité de Génération de Codes Dirigée par les Modèles. Journées nationales du GDR GPL, Pau, France. March 2010

Malenfant, J. 2002. Modélisation de la sémantique formelle des langages de programmation en UML et OCL. *Writing*.

Marcano, R., Levy, N.2002. Transformation rules of OCL constraints into B formal expressions. Technical report.

Muller, P. and al. 2006. Model-Driven Analysis and Synthesis of Concrete Syntax. *Model Driven Engineering Languages and Systems*, 2006, pp. 98-110

Nemo, C., Blay-Fornarino, M. , Riveill, M. 2008. Multi-application de patterns. In *IDM'2008* 5-6 june, Mulhouse.

Paulin- Mohring, C. and al. 2009. Cours 2 : Exemple de développement Gallina : Sémantique d'un mini-langage http://www.lri.fr/~paulin/MPRI/notes/index-2-7-2.html

Sen, S., and al, 2009. Meta-model pruning. In *ACM/IEEE 12th International Conference on Model Driven Engineering Languages and Systems (MODELS'09),* Denver Colorado.

Wimmer, M. and Kramler, G. 2005. Bridging Grammarware and Modelware. *Satellite Events at the MoDELS 2005 Conference,* pp. 159-168

Copy Rights

GENERAL MODELLING APPROACH BASED ON THE INTENSIVE USE OF ARCHITECTURAL AND DESIGN PATTERNS

Anna Medve*, László Kozma** and Ileana Ober***
*PhD School of Computer Science, Eötvös Loránd University, 1117 Budapest, Hungary, Pázmány Péter sétány 1/C
**Department of Software Technology, Eötvös Loránd University, 1117 Budapest, Hungary, Pázmány Péter sétány 1/C
***IRIT- Université Paul Sabatier Toulouse, 31062 Toulouse, France,118, route de Narbonne

ABSTRACT

This paper reports on general modelling approach based on the intensive use of architectural and design patterns. This method addresses the pair-wise direct mapping of service factories from the problem domain to POSA2 architectural patterns features. Architectural pattern languages (functionally interrelated architectural patterns) intent and scope help to define the boundaries and context of the system, and the identification of archetypes to decompose the system into its main components. Modelling behavioural sequences and architecture-level model transformations by weaving GoF patterns are proposed. The use of UML2.0 Composition and Sequence diagrams for the definition of pattern properties in addition to textual descriptions of POSA2 patterns are proposed. This paper reports on work in progress and the heuristics we provide here, could serve as basis to define a methodology for modelling quality levels of models reuse.

KEYWORDS

Design Patterns, POSA2, MDE, UML2.0, Pattern-based, Software Reuse

Copy Rights

1. INTRODUCTION

The ability to react rapidly to changing market conditions and user requirements is the most important success factor in today's software industry. The design patterns are better features for reuse. The better use of the know-how of experts can reduce efforts and can improve the quality of software.

In the developer's community, several pattern-based methods have been proposed to enable the intensive reusing in component-based development. Several methods aim at the integration of pattern systems into existing methodologies to raise reuse and quality (Eriksson et al., 2000), (Gaudin et al, 2007), (Stal, 2006). A common aspect of most of these methods is the strong focus on pattern implementation. As a result, in earlier phases of the software process the design patterns are not well adapted to several analysis and classification tasks, and the pattern-based modelling of quality also not.

The paper proposes a general UML2.0 (Unified Modelling Language) (OMG, 2010) modelling approach base on the interlocking of architectural patterns with design patterns to reduce the level of abstraction to create an executable representation of the system and its parts in the context of network-centric systems.

The work we present here is a natural continuation of our previous works (Medve and Ober, 2008), in an attempt to fill the gap between models and their implementations by software abstractions in a MDA spirit. As explained at Ober et al., (Ober et al., 2010), the full value of MDA is only achieved when the modelling concepts map directly to domain concepts rather than computer technology concepts.

We introduce the use of Design Patterns (GoF) (Gamma et al, 1995) by weaving with Pattern-oriented Software Architecture of Networked Systems (POSA2) (Schmidt et al., 2000) architectural pattern language in modelling at Requirement modelling level to better assure the conformance of Architectural models to the Requirements models and to the Implementation Models, also. The rest of this paper is organized as follows: Section 2 exposes our approach, its application on the development of a case study that we use to validate our ideas. The paper ends with a conclusion giving directions for future work.

2. GENERAL MODELLING METHOD ON INTENSIVE USE OF PATTERNS

Software architecture is a reusable, transferable abstraction of a system. Architectures should be described by a set of views that supports its analysis and communication needs (Clements and Kazman, 2001). Reference architecture is a set of domain concepts mapped onto a standard set of software components and relationships.

POSA2 (Pattern-oriented Software Architecture of Networked Systems) architectural patterns (Schmidt et al., 2000) help to resolve common design challenges of distributed systems. POSA2 forms the group of patterns Service Access and Configuration Patterns, Event handling Patterns, Synchronization Patterns and Concurrency Patterns. These groups of patterns are interrelated and form an architectural pattern language. In our approach, POSA2 provide guidance in design choices earlier during the requirements elicitation phase and ask the questions: when, who, what, where, how, strong related on domain context.

The UML2.0 allows the creation of multi-view modelling method techniques with the combinations of available analysis and design techniques. The UML2.0 composition diagram (see at Figure 3) allows to represent patterns either individually or to integrate them into the functional parts. It corresponds to a pattern or weaved patterns specification, forming a reusable pool of weaved patterns. The classes are architectural units for subsystem functionalities. The interactions with the environment and the other parts of the system are modelled by ports. The communication units are modelled by the required and provided interfaces, which are considered to be the operations of functions. We analyze and finalize the dynamicity of functions by UML2.0 Sequence diagramming models of communicating units.

The general modelling method on intensive use of patterns consists of two main phases: creating the reference architecture model on domain context, and search and weave among them the GoF patterns. The main steps are: First, the structure of the abstract mechanisms given by the design patterns is instantiated in UML Composition Diagram (Step 1). Second, the collaborations of structural elements are decomposed using UML Sequence Diagrams (Step 2). Third, the decomposed collaborations offer the act of recovering GoF design patterns as communication abstractions (Step 3). Finally, after several refinements there are several ways to create executable forms of the target, i.e. through UML State-machine Diagrams (Step 4).

To report on this method, in this paper we focus on the Component Configurator pattern from POSA2 Service Access and Configuration Patterns group (Schmidt et al., 2000). The Component Configurator pattern allows an application to link and unlink its component implementations at run-time without having to modify, recompile, or statically re-link the application. Its corresponding UML2.0 Composition diagram is shown in Figure 1 and the Sequence diagrams in Figure 2. The roles in the Component Configurator pattern are to define a unique interface to configure and control operations. The ConcreteComponents implement the component control interface to provide a specific type of component, the ComponentConfigurator coordinates the (re)configuration of concrete components and the ComponentRepository manages all concrete components.

The model consists of several diagrams from the following steps of parallel composition of structural and behavioural views:

Step 1. *Instantiation of roles:* Using software abstractions and textual descriptions that are contained in POSA2 book take the roles from collaborating objects and events. Find the communicating instances of the pattern, and create corresponding classes in the architecture diagram. In our example this leads to the creation of Configurator, Component and Repository instances of the pattern. This leads to the creation of Configurator, Component and Repository features in Composition Diagram and in Sequence Diagram (Figure 1).

Step 2. *Initiate the operations of instantiated roles:* This initiates the configuring and controlling events by a set of messages, such as Init(), Suspend(), Resume(), Fini(), Insert(), Run() and Remove(). The order of these events depends on the causal relationships existing in textual descriptions of the context of Component Configurator pattern (Schmidt et al., 2000) and it is established during the context analysis. Figure 1 shows the obtained reference architecture. Figure 2 shows the corresponding sequence diagram of the scenario for component links actions at the level of detail of implementations decisions.

Copyright

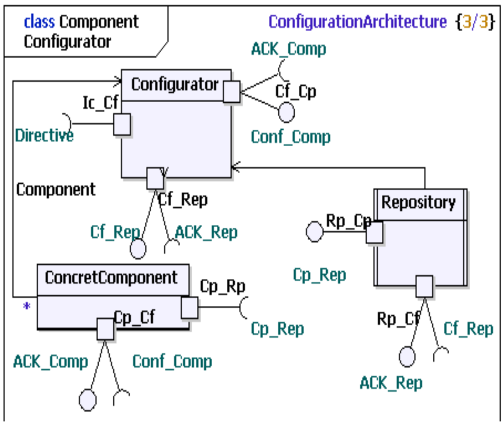


Figure 1. The reference architecture of component configurator

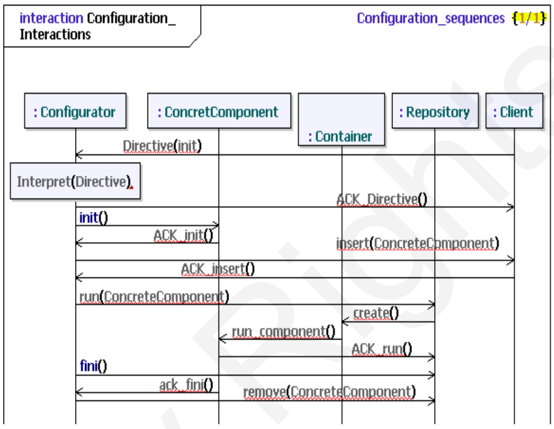


Figure 2. Behavioural sequence and architecture transformation

Step 3. *Identify a GoF pattern:* during the analysis of operations in the context of architectural pattern. In our case, see Figure 2 the Init() event details are reduced onto the application of the Interpret pattern plus two way hand check messages (see Figure 2). We identify the Memento and the Interpret GoF patterns, as methods that implement state() and directive() operations, while the Observator() pattern can implement the Notifier() operation.

Step 4. *Refine the architecture from analysis results:* this step consists of iterations for the completion of CompConfig composition diagram (see Figure 1) by required and provided interfaces.

Copy Rights

2.1 Example

In this section, we consider the example of a simplified GSM (Global System for Mobile Communications) network (ETSI, 2000) which is classified as client–server communication model. The behaviour of a GMSC (Gateway Mobile Switching Centre) call management process consists of initiating a call received from a local or a remote mobile station, handling the occurred errors, ensuring the correct performance of connection, and ending the call. We use the above presented generic model of services access and configurations architectural patterns' intent and scope to select the service functionalities of features in the domain context. In the GSM domain, the GMSC unit is matched by Component Configurator pattern.

Figure 3 shows the GMSC unit functionalities matched by its role of service configuration using the Component Configurator reference architecture. The events of roles uncovered by the pattern are introduced on architectural level and they are further detailed during the analysis for detailed design, performed with communication sequences.

The analysis of GMSC context consists of classifying the GMSC actions as server functionalities with sub-classes and operations and defining the methods that implement the operations. This is done by applying the reference architecture of the pattern Component Configurator and by direct mapping their roles to functionalities of GMSC unit. Their mapping is based on the roles established within the architecture model of GMSC and mapping is easy for novice also based on sequence diagram of generic reference model see at Figure 2. The GMSC events are classified as operations of roles (instances) from the pattern. Moreover, we define the methods implemented by the operations. For example, the callHandler responsibility of GMSC is covered by the roles of Concrete Component instance. In the next step, we map directly the roles and obtain the model of context behaviour, on which we apply the Reference architecture of Component Configurator (see Figure 1) for detailed design of behaviour.

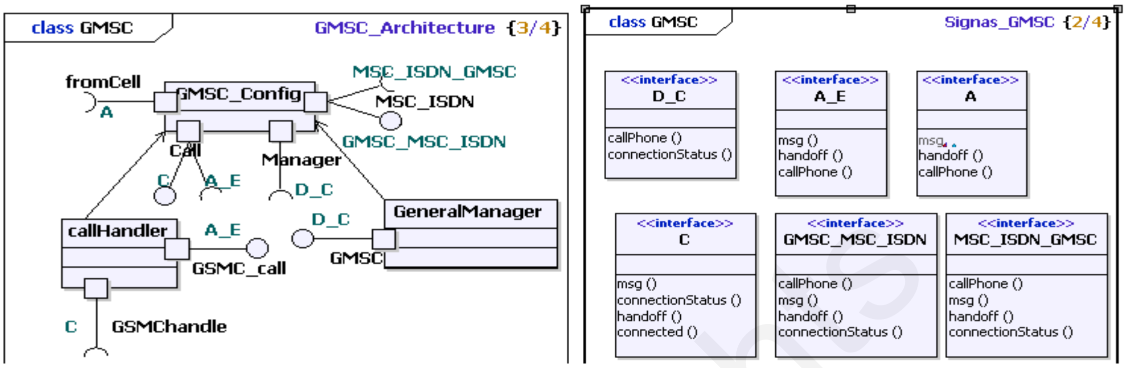


Figure 3. The architecture of GMSC matching from the reference architecture of component configurator

3. CONCLUSION

The general modelling approach for pattern-based reference architecture modelling for networked distributions, are discussed. The use of UML2.0 Composition and Sequence diagrams for modelling behavioural sequences and architecture transformations with GoF patterns are proposed. The novelty in this approach is the use of scenarios and remote call with some inline operator to weaving the discovered GoF patterns into POSA2 patterns. This constitutes a vertical model transformation conducted by implementation context of POSA2 (Schmidt et al., 2000) architectural patterns useful in MDE model transformations. Applying any design pattern remains quite complex and requires a good knowledge of rather important set of concepts that are reused to better support traceability between requirements and implementation models. Future work will be extend our method by some consistency checking (Medve and Kozma, 2010).

Copy Rights

ACKNOWLEDGEMENTS

This research work was supported by TÁMOP-4.2.1/B-09/1/KMR-2010-0003.

REFERENCES

Clements, P. and Kazman R., 2001. Evaluating Software Architectures: Methods and Case Studies. Addison-Wesley

Gamma, E. et al., 1995. *Design Patterns: Elements of Reusable Object-Oriented Software.* Addison-Wesley.

Gaudin, E. et al, 2007. Design for Dependable Systems,*13th International SDL Forum*, Paris, France, LNCS 4745.

Eriksson, H.E. et al, 2000. *Business Modeling with UML Business Patterns at Work*. Wiley.

ETSI, 2000. GSM 01.01 version 8.0.0, Release 1999, TS 101 855 DTS/SMG-000101Q8, ETSI.

Medve, A. and Ober, I., 2008. From Models to Components: Filling the Gap with SDL Macro-patterns IEEE CIMCA Innovation in Software Engineering, December 9-12, 2008, Wien, Austria, pp.1252-1257.

Medve, A. and Kozma, L., 2010. MDA process and model refinements based on VERIMAG IF verification tools, 8th Joint Conference on Mathematics and Computer Science, July 14-17, 2010, Komárno, Slovakia (to be published)

Ober, I., et al., 2010. Dealing with variability within a family of domain-specific languages: comparative analysis of different techniques. *Innovations System Software Engineering*. Vol. 6, pp 21–28.

OMG 2010. UML2.0, MDA/MDE standards, www.omg.org

Schmidt, D. et al., 2000. *Pattern-Oriented Software Architecture-Patterns for Concurrent and Networked Objects*. Wiley.

Stal, M., 2006. Using Architectural Patterns and Blueprints for Service Oriented Architecture. In:"Future Trends of Software Architecture", *IEEE Software* 23/2 pp.54-61.

Copy Rights

A COMPUTATION EVALUATION OF SOME SOFTWARE FOR MATHEMATICAL PROGRAMMING

Themistoklis Glavelis*, Nikolaos Ploskas* and Nikolaos Samaras**
PhD Candidate/ Assistant Professor**, Department of Applied Informatics, University of Macedonia*
156 Egnatia Str., 54006 Thessaloniki, Greece

ABSTRACT

The purpose of this paper is to present some of the most well-known commercial and open source software for mathematical computer programming. Specifically, the commercial programs which will be presented are Maple, Mathematica and Matlab and the open source software are Octave and Scilab. These software packages are mainly used by scientists and software engineers. The strengths and weaknesses of these programs are detailed described through the chapters of this paper. Finally, there is a speed comparison of some frequently used algorithms, procedures and operations which are build-in functions of these mathematical programming environments.

KEYWORDS

Scientific Computing, Computational Comparison, Mathematical Software.

Copy Rights

1. INTRODUCTION

The aim of this paper is to present a description and comparison of some of the most popular mathematical software. Maple (http://www.maplesoft.com/), Mathematica (http://www.wolfram.com/) and Matlab (http://www.mathworks.com/) are selected from other commercial software due to their wide spread among the scientific community and industry. Furthermore, they all belong to the leaders of software enterprises in mathematical programming. Moreover, Octave (http://www.gnu.org/software/octave/) and Scilab (http://www.scilab.org/) are picked because they are the most well-known open source software for mathematical programming. All of the reviewed software is very large programs and this paper does not even attempt to cover their scope. There have been many reviews of each system individually, but only a few comparisons between them that are either obsolete or don't compare all these five software (Kendrick & Amman, 1999), (Shacham et al, 1998), (Zotos, 2007), (Alsberg, 2006). The main advantage of these scientific computing environments is their ability to be used either from experienced software engineers or from amateur programmers.

Maple is a great choice for engineers, mathematicians, and scientists due to the fact that it is the result of over 25 years of cutting-edge research and development under the Waterloo Maple Inc. (also known as Maplesoft). Furthermore, there is a large number of tutorials and books referring to Maple utilities (Garvan, 2001), (Parlar, 2000). Apart from that, Maple has been selected from a vast number of universities world-wide to be used at the frameworks of lessons like mathematics. The latest compiled version of Maple is Maple 13. Mathematica is a computational mainly software program used in scientific, engineering, and mathematical fields which is developed and distributed by Wolfram Research which was founded by Stephen Wolfram in 1987. Moreover, there are many articles and books which describe analytically the advantages and weaknesses of Mathematica (Dubin, 2003), (Tott, 2004). The latest compiled version of Mathematica is Mathematica 7. Matlab is a matrix language developed and distributed by the MathWorks Inc. As the name suggests, Matlab (MATrix LABoratory) is especially designed for matrix computations like, solving systems of linear equations or factoring matrices. Apart from that, it is general accepted that MATLAB and its numerous toolboxes can replace or enhance the usage of traditional simulation tools for advanced engineering applications. All these toolboxes and functions have thoroughly described in a numerous tutorials and guides (Palm, 2008), (Colgren, 2007). The latest compiled version of Matlab is Matlab R2010a. Octave is open source software where the users are able to enhance and adapt the source code according to their demands. Consequently, everyone is free to use it and free to redistribute it on certain conditions. Octave is a high-level language, primarily intended for numerical computations. Due to the fact that Octave is an open source software, there are many documentations either as books or available on-line through the internet (Eddelbuettel, 2000), (Malcolm, 1997). The latest compiled version of Octave is 3.2.3. Scilab is a scientific software package for numerical computations providing a powerful open computing environment for engineering and scientific applications. Scilab is open source software and since 1994 it has been distributed freely along with the source code via the internet. Moreover, Scilab includes a large number of functions and toolboxes which have been many times described in a large number of books and tutorials (Campbell, 2006), (Zhang, 2006), (Mrkaic, 2001). The latest compiled version of Scilab is 5.2.1.

An outline of the rest of the paper is as follows. In section 2, general descriptions of the systems are given. Furthermore, we introduce the numerical libraries and the useful packages and toolboxes of each software. Finally, a speed comparison of some frequently used procedures within mathematical models is presented in section 3. Finally, in section 4 we outline our conclusions.

2. NUMERICAL LIBRARIES

First of all, it is general accepted that numerical libraries is a sector of great value for mathematical programming languages. All reviewed programs have a big range of built-in mathematical functions and procedures. Apart from that, the reviewed software is not constrained only to mathematics but they are spread among many other scientific fields, like statistics and econometrics.

As it is mentioned previously, all reviewed software has many and adequate functions for most linear algebra calculations. These built-in functions have been implemented in order to simplify the process of mathematical programming for experienced and amateur users. All functions are parts of extensive numerical algorithms for a wide range of applications. Maple and Matlab are the most adequate software for linear algebra, while Mathematica, Octave and Scilab lack some functions like Smith Normal Form.

Another significant category of functions is the numerical analysis, a wide spread mathematical tool among the industry and the scientific community. Maple, Mathematica and Matlab have the most functions for numerical analysis. In contrast, Scilab lacks a function for k-Spline Interpolation and Octave misses some functions, like Inverse Fourier Transformation, Bisection and Newton method.

Statistics is a powerful tool for engineers, scientists, researchers and financial analysts in order to collect, analyze explain, and present their data. Consequently, statistics is also a significant section for mathematical programming languages. All reviewed software is an ideal option for statisticians, because they support a wide range of tasks, from basic descriptive statistics to developing and visualizing multidimensional non-linear models. They offer a rich set of statistical plot types and interactive graphics, such as polynomial plotting and response surface modelling.

Econometrics combines economic theory with statistics to analyze and test economic associations. Theoretical econometrics considers questions about the statistical properties of estimators and tests, while applied econometrics is concerned with the application of econometric methods to assess economic theories. Economical data are generally observational, rather than being derived from controlled experiments. Early work in econometrics focused on time-series data, but now econometrics also fully covers cross-sectional and panel data. For all these reasons, econometrics is a very important issue for every mathematical program. Mathematica and Matlab are more adequate in the area of econometrics, because they have more built-in functions which can be used from an econometrician than the other software. Maple and Octave lack some functions, like Durbin-Watson Test, and finally Scilab includes only the primary functions for econometrics.

Copyrights

3. SPEED COMPARISON

In order to gain an insight into the practical behavior of each one of the reviewed software, some computational experiments are proposed. The computational comparison has been performed on an Intel Pentium IV processor with 3.4 GHz and 1 GB RAM running under Windows 7. The reported CPU times were measured in seconds. The speed comparison tests 25 functions which are very often used within mathematical models and not only. Test set problems are categorized into five groups: miscellaneous operations, matrix operations, basic algebra, advanced algebra and statistics. The results given in Table 1 show the average times of 10 executions. All runs were made as a batch job. The 'winning' time for each test is given in bold.

Table 1. Speed comparison.

TEST	Maple (13)	Mathematica (7)	Matlab (R2008b)	Octave (3.2.3)	Scilab (5.2.1)
MISCELLANEOUS OPERATIONS					
Loop test 100000 x 100000	11930,598	31819,800	**9774,985**	43425,000	72147,885
3000 x 3000 random matrix ^ 1000	4,960	**0,671**	4,103	3,463	4,353
Sorting of 10000000 random values	0,950	0,702	**0,267**	0,296	0,328
FFT over 2^20 random values	0,857	0,515	0,614	**0,484**	2,013
Calculation of 2000000 Fibonacci numbers	198,464	19,718	**2,066**	4,571	4,680
Plot 2d on 200000 points	0,953	**0,134**	1,560	1,243	1,076
Plot 3d on 200000 points	2,745	**0,456**	4,876	2,564	10,085
Average performance for the tests of this group	30,24%	**67,59%**	59,01%	43,66%	27,92%
MATRIX OPERATIONS					
Matrix multiplication among two 3000x3000 random arrays	14,760	10,840	**9,504**	13,276	17,847
Transpose of a 3000x3000 random matrix	1,160	0,172	0,149	**0,125**	0,359
Hessenberg form of a 3000x3000 random array	167,510	49,077	**36,915**	174,130	41,683
Average performance for the tests of this group	32,40%	78,54%	**94,63%**	64,26%	58,86%
BASIC ALGEBRA					
Determinant of a 3000x3000 random array	5,700	6,848	4,537	5,616	**4,336**
Inverse of a 3000x4000 random array	37,050	19,984	**12,707**	16,224	14,274
Eigenvalues of a 3000x3000 random array	390,530	**62,853**	407,302	487,500	171,850
Eigenvectors over a 3000x3000 random array	945,490	293,663	**133,780**	1039,000	1406,530
3000x3000 dot product matrix	12,870	12,652	**5,885**	6,895	10,202
Linear system solve of 3000 equations	5,880	5,678	**4,950**	5,679	163,410
Average performance for the tests of this group	45,67%	68,18%	**85,91%**	59,56%	50,08%
ADVANCED ALGEBRA					
Cholesky decomposition of a 2000x2000 random array	5,670	**2,010**	2,485	2,683	2,262
Lu decomposition of a 1500x1500 random array	7,960	5,897	**4,701**	5,834	20,286
Qr decomposition of a 1200x1200 random array	121,870	19,640	**17,469**	21,263	17,909
Singular value decomposition of a 2000x2000 random array	1015,640	93,054	**20,349**	576,950	52,915
Schur decomposition of a 1500x1500 random array	1626,790	27,050	**4,519**	733,790	95,566
Average performance for the tests of this group	22,23%	61,45%	**96,18%**	48,36%	50,55%
STATISTICS					
Principal component factorization over a 3000x300 random array	95,645	8,740	8,375	**6,540**	7,830
Gamma function on a 3000x3000 random matrix	6,380	7,085	**0,855**	30,092	3,885
Gaussian error function on a 3000x3000 random array	1,930	9,623	1,300	**0,187**	0,453
Linear regression over a 3000x3000 random array	60,150	4,852	4,798	5,710	**2,730**
Average performance for the tests of this group	8,62%	36,28%	62,35%	**62,66%**	61,71%
OVERALL PERFORMANCE	27,83%	62,41%	**79,61%**	55,70%	49,82%

Copy Rights

The software's performance for each test has been calculated using the relation

$$\frac{BT < software >_k}{T < software >_i} x(100), i = 1, 2, ..., 5, i \neq k$$

where $BT < software >_k$ is the best timing result for a specific test, run by all five programs and $T < software >_i$ is the timing result of the software i, i ≠ k for the same test. The best timing is then set equal to 100%. To calculate the overall performance for each one software, we add the percentage values for every test and divide it by the total number of tests. Finally, the largest percentage corresponds to the best overall performance.

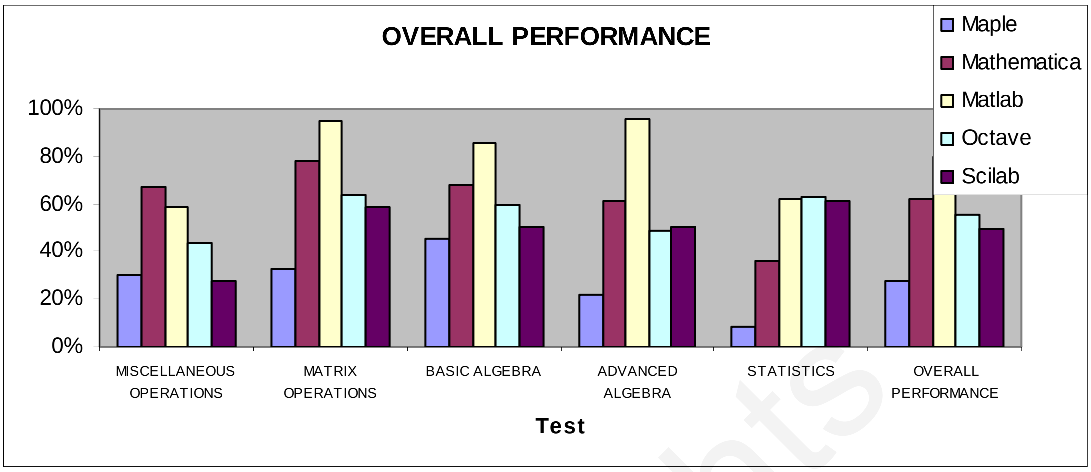


Figure 1. Group and overall performance

In the above figure, the average time of each group of tests are presented for all reviewed software. The first group refers to miscellaneous operations like 2d and 3d plotting, sorting numbers. In this group Mathematica is the winner and Matlab follows with no significant difference. On the other hand, Scilab has the worst time which mainly depends on the loop test and 3d plotting. In the next group test which refers to matrix operations, Matlab has the best performance. In contrast to Matlab, Maple demands more time to accomplish the matrix operations and it came to the last position of these tests. In basic algebra, Matlab is again the winner and Mathematica comes next. The other three programming languages have no significant differences. In the next group of tests, advanced algebra, Matlab has the most significant difference comparing to execution times of the other software. Moreover, Octave and Scilab have almost the same performance and Maple comes last. In the last group of tests, which refers to statistics, Matlab, Octave and Scilab are very close with Octave to lead the average of execution times. Finally, in overall performance Matlab includes the best results due to the fact that it comes first in three of the five groups of tests.

Copy Rights

4. CONCLUSIONS

The choice of data analysis software is not an easy decision and it depends on the needs and expectations of users. Maple, Mathematica, Matlab, Octave and Scilab are easy-to-use languages which allow a fast implementation and prototyping of mathematical and statistical algorithms. Despite their strong similarities, there are substantial differences between these matrix programming languages. They differ from each other in terms of usability, richness and finally in terms of performance.

Maple is the best solution for doing symbolic computations. Moreover, Maple has a large library of mathematical and econometrical functions. The main disadvantage of Maple is its' speed.

Mathematica is generally regarded as having the best graphical capability. Furthermore, Mathematica has a wide variety of functions for almost everything. A real weakness of Mathematica is its' complex syntax.

Matlab's strengths are the large number of available toolboxes, the possibility to develop graphical user-interfaces easily and the functions and competence for nearly all important topics. In contrast, the symbolic calculations with Matlab are weak.

Octave is a language compatible with Matlab, but it is not a top performer on Windows. It has a wide variety of mathematical and statistical functions that makes it a powerful tool for numerical computations.

Scilab is a free alternative of Matlab. Scilab offers a large number of functions for mathematical programming. Its performance at speed comparison is quite satisfactory and one disadvantage of Scilab was the restrictions referring to the dimensions of data that can create and handle.

REFERENCES

Alsberg B. and Hagen O. J., 2006. How can octave replace Matlab in chemometrics. *Chemometrics and Intelligent Laboratory Systems*. Vol. 84, No. 1-2, pp 195-200.

Campbell S. et al., 2006. *Modeling and simulation in Scilab/Scicos*, Springer, New York. USA.

Colgren R. D., 2007, *Basic Matlab*, Simulink and Statefow. AIAA Education Series, Virginia, USA.

Dubin D., 2003. *Numerical and Analytical Methods for Scientists and Engineers*, John Wiley & Sons Inc,

Eddelbuettel D., 2000. Econometrics with Octave. *Journal Of Applied Econometrics*, Vol. 15, pp 531-542.

Garvan F., 2002. *The Maple Book*. CRC Press, Florida, USA.

Kendrick, D. A. and Amman, H. M., 1999. Programming Languages in Economics. *Computational Economics*, Vol. 14, No. 1-2, pp 151-181.

Malcolm M., 1997. Octave: A Free High-Level Language for Mathematics. *Linux Journal*, Vol. 1997 (July), No.8.

Mrkaic M., 2001. Scilab as an Econometric Programming System. *Journal Of Applied Econometrics*, Vol. 16, No. 4, pp 553–559.

Palm W. J., 2008. *A Concise Introduction to Matlab*. McGraw-Hill, Rhode Island, USA.

Parlar M., 2000. *Interactive Operations Research with Maple*. Birkhauser, New York, USA.

Shacham M. et al, 1998. Comparing for interactive solution of nonlinear algebraic equations. *Computers & Chemical Engineering*, Vol. 22, No. 1-2, pp 323-331.

Tott M., 2004. *The Mathematica GuideBook for Programming*. Springer/Verlag, New York, USA.

Zhang Z., et al., 2006. *Lecture Notes in Computer Science*, Springer, Berlin Heidelberg, Germany.

Zotos K., 2007. Performance comparison of Maple and Mathematica. *Applied Mathematics and Computation*, Vol. 188, No. 2, pp 1426-1429.

(2010) The Maplesoft website. [Online]. Available at: http://www.maplesoft.com/

(2010) The Wolfram Research website. [Online]. Available at: http://www.wolfram.com/

(2010) The MathWorks website. [Online]. Available at: http://www.mathworks.com/

(2010) The John W. Eaton. website. [Online]. Available at: http://www.gnu.org/software/octave/

(2010) The INRIA website. [Online]. Available at : http://www.scilab.org/

Copy Rights

Reflection Papers

Copy Rights

Copy Rights

MOBILE COMMUNICATION APPLICATION FRAMEWORK FOR HEALTH CARE

Toshiyuki Maeda*, Yuki Ando**, Yae Fukushige*** and Takayuki Asada**
**Faculty of Management Information, Hannan University, Japan*
***Graduate School of Economics, Osaka University, Japan*
****Graduate School of Commerce, Otaru University of Commerce, Japan*

ABSTRACT

This paper presents a software development framework for Web/mail applications, based on framework known as MVC. In our study, one M and one C are prepared and several Vs are developed for each interface as Web, e-mails, and so on. We introduce an application for health care education at universities using this framework, and discuss effects

KEYWORDS

Mobile communication, health care, application framework, mobile phone, MVC.

Copyrights

1. INTRODUCTION

Recently mobile communication systems (MCSs) are getting more and more important for our everyday life as we cannot live without mobile phones, not only for business but also private time. So far, there are several researches (for instance [MTA04, MOFA09]), especially for learning support management.

In various education areas, many problems are solved using the web-based systems [HGMA04]. One of the most critical problems is, however, that web-based systems cannot be used in lecture rooms where computers are not settled for all students, and is essential for many cases. We have thus developed an e-mail-based system using mobile phones, which almost all students have in Japan. There are only few e-mail-based systems for similar purpose, such as [JvRS96].

We furthermore have an attempt to apply for health care support. For that purpose, it is important to be easy not only to reconfigure but also maintain and improve MCS. We therefore introduce software development environment, based on MVC framework (such as [rub, cak] etc.), where M stands for Model (information structure), V for View (input and output as user interface), and C for Controller (data control and processing). In our study, one M and one C are prepared and several Vs are developed for each interface such as Web, e-mails, and so on. In this paper, we explain conventional system and later describe the proposal of Web/Mail application framework. Lastly we show some experimental results and discussion.

2. HEALTH CARE EDUCATION

Our main objectives for health care education are below two;

- preventing health disorder, and
- improving health literacy of students.

Hence we have several concrete targets for students;

- understanding their own situation objectively,
- learning preferable behaviors for preventing overweight,
- enabling to set their tangible goals,
- putting the knowledge to practical use,
- continuing to behave for preventing overweight,
- reducing BMI,

- understanding necessity of self-management.

We thus propose health care system regarding to the conventional system as follows;

- System must be improved health care education (mainly out of class),
- System has no concept of attendance (used in everyday life), and
- System supports continual communication (no need for real-time response).

3. FRAMEWORK FOR WEB/MAIL APPLICATIONS

In this paper, "Mail application" denotes an application where sending mails from users to a server is treated as input and replies from the server to corresponding users are as output, and repeating this interaction (round-trip) makes a set of communication for the application system. In general, MCS has three layers structure as for users;

1. System-Administrator (SA),
2. Intermediate-Administrator (IA) : teacher, doctor, nurse, etc.,
3. End-User (EU): students, patients, etc.

For EUs, not WWW but e-mails are seemed to be essential for our MCS, which is because;

1. carrier- or terminal-independent; as some WWW browsers on mobile phones are not compatible (HTML nor Java applet) with others, especially in Japan, and so we have to provide several variation of application for those phones, though e-mails are compatible with each other.
2. lightweight process; as WWW applications are heavy for servers in general.
3. quick response; though some exemption may happen.

It seems, however, to be convenient for not only SAs but also IAs, who know not so much about internal structure of the MCS.

Copy Rights

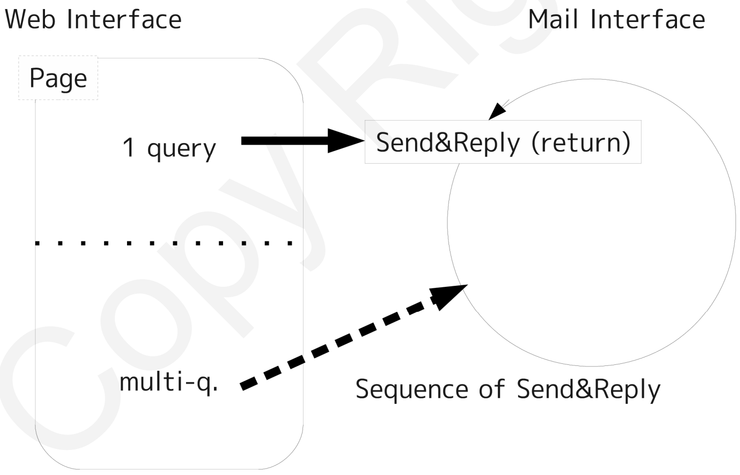


Figure 1. Web/mail interface.

Figure 1 shows procedures of Web and Mail interfaces. For a web application, HTML FORM is used for input to an application. We can send multiple input data simultaneously on web application, though it is not so easy for mail application

4. PROTOTYPE AND FIELD TEST EXPERIMENT

The features of this system concept are to be able to collect, accumulate, process and distribute information anytime and anywhere at low cost using mobile phone. We proposed our previous system in a broad range of applications in the medical and health care fields [MOFA08, MOFA09]. Although several different applications were proposed, these had some institutional and practical problems that were difficult to resolve immediately. To avoid these problems and conduct an experimental trial, we apply this system in changing the lifestyle habits of obese students through diet and exercise monitoring. For this trial, which took place in 2009, we have developed a new prototype system by adapting, and worked closely with university public health nurses.

We have had field tests as three experimental trials. The first and most important part is daily questionnaire for students' food intake, physical activity, and lifestyle habits. The questionnaire is sent and received through the questionnaire function using mobile e-mail communication. The second part is provision of information via message distribution function. The third part is individualized instruction message to a student when the health nurse deemed it necessary. Details and results of the field tests are as follows:

- 31 test subject students (20 male, 11 female) from April until July in 2009.
- -0.93kg (average), 2.83 (standard deviation)
- From -12 to +4 kg variation

Though this trials have not achieved tremendous effects on student lifestyle habits, this trial also revealed some problems: one of negative results is the low motivation of students, which indicated in the return rate of daily questionnaire tends to decline in process of time. Figure 2 shows return rate of this experiment period.

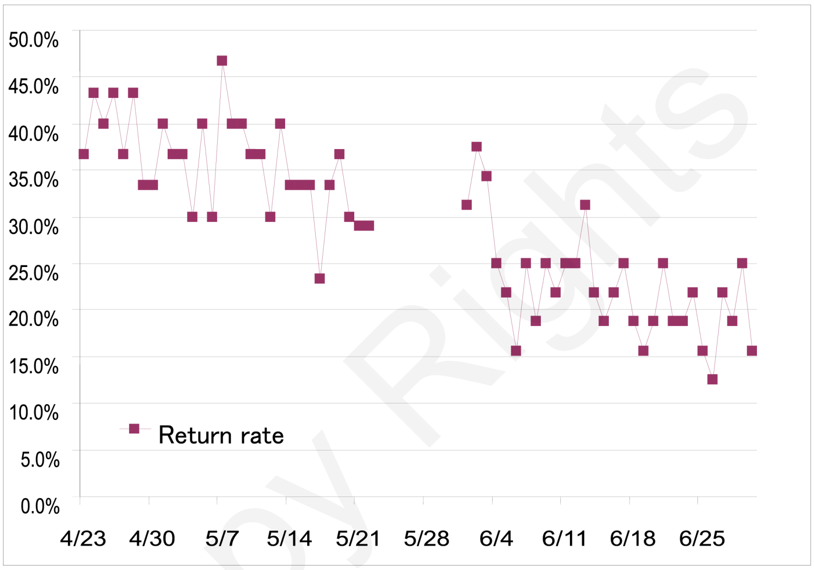


Figure 2. Return rate.

Copy Rights

Note that since 5/23 until 5/30 we could not precede the system as the university had been temporally closing because of the outbreak of swine flu. We regarded the low motivation as caused mainly two factors. One of the reasons is seemed the attitudes of students, as many students have not realized the necessity of changing lifestyle habits, and there are no rewards for this trial. Another one is thought as input burden, where there may be an intrinsic problem for using of mobile phone as interface, which is suggested the average numbers of input characters as around 60.

5. CONCLUSION

This paper presents development framework for Web/Mail multi-application, prototype implementation and some experimental results. For the time being qualitative and/or quantitative evaluation is not examined enough and so we believe we must do any other examination for that. In future work, with view of usability, bulk mail interface should be reconsidered essentially.

ACKNOWLEDGEMENT

This research was partially supported by the Ministry of Education, Science, Sports and Culture, Japan; Grant-in-Aid for Scientific Research (A), 19201032. Part of this study is also conducted under a sponsorship of Grant of Institute of Industrial and Economic Research, Hannan University. The authors greatly appreciate the supports.

REFERENCES

[cak] CakePHP: the rapid development php framework. http://cakephp.org/.

[HGMA04] N. Hanakawa, K. Goto, T. Maeda, and Y. Akazawa. "discovery learning for software engineering -a web based total education system: Hint-". In "Proceedings of the International Conference on Computers in Education in 2004", pages 1929-1939, Melbourne, Australia, 12 2004.

[JvRS96] Dag Johansen, Robbert van Renesse, and Fred B. Schneider. Supporting Broad Internet Access to TACOMA. In Proceedings of the 7th SIGOPS European Workshop, pages 55-58, Connemara, Ireland, 1996.

[MOFA08] T. Maeda, T. Okamoto, Y. Fukushige, and T. Asada. Learning Session Management With E-mail Communication. In Proceedings of World Conference on Educational Multimedia, Hypermedia & Telecommunications (ED-MEDIA 2008), pages 1787-1792, Vienna (Austria), 2008.

[MOFA09] T. Maeda, T. Okamoto, Y. Fukushige, and T. Asada. Learning Session Management With E-mail Communication. In Proceedings of World Conference on Educational Multimedia, Hypermedia & Telecommunications (ED-MEDIA 2009), pages 1156-1161, Honolulu (HI, USA), 2009.

[MTA04] T. Maeda, M. Tomo, and T. Asada. Integrated lecture-support system using e-mail. In Proceedings of National University Symposium on Information Edication Methods (in Japanese), pages 26-27, Tokyo (Japan), 7 2004.

[rub] Ruby on Rails. http://rubyonrails.org/..

Copy Rights

TRACKING ELECTRONIC DOCUMENTS IN ORGANIZATIONS

Majed AbuSafiya
Faculty of Information Technology
Alahliyya Amman University

ABSTRACT

Electronic documents are frequently created within organizations for communication, coordination and as part of business processes. These documents are usually stored within the file systems of their authors' personal computers. These documents are not easily located or queried whenever needed. In this paper, a software solution is proposed where these documents are automatically and transparently collected from the personal computers in the organization and maintained in a database management system for automated queriability and retrieval. This solution has the advantage over document management systems where no explicit document upload or management overload is required from the employees.

KEYWORDS

Document Management Systems, Organization, Database management System, RMI

Copy Rights

1. INTRODUCTION

Employees within any organization frequently create different types of electronic documents using different software installed in their personal computers (e.g. emails, memos and reports). Documents could also be generated using very specific software (e.g. a Java program created by a programmer or an AUTOCAD file generated by an engineer). Many activities of the business processes are associated with documents such that a document can be considered as proof that a certain activity took place. Documents are also used to document deviations from business processes and for communication and coordination. A document could also be the final deliverable of a business process.

The question we want to answer is: how can we have a better management solution of these documents so that they can be queried in automated manner? This solution should be easy to implement, it should consider all types of electronic documents created by the employee, it does not compromise the flexibility in working with such documents, and does not add extra workload (on the employee) to manage them. The proposed solution is based on finding newly created/modified documents. These documents are then transferred to be stored (with all important attributes) in a database management system that provides the desired automated queriability.

The solution we propose considers two main important categories of documents. The first category is emails. Emails are maintained in a database management system by frequently checking the email accounts of the employees for recently received emails and saving them in the document database. The second category is documents that are created by the employee using different software systems installed on his personal computer (e.g. MSOffice). These documents are collected from the personal computers of the employees and stored in the document database. This can be achieved through a client-server solution where a client software is installed in the employees' machines. This client software is scheduled to run periodically to traverse the file system of the hosting machine to extract those documents that were recently created or modified after the last traversal. Once found, these documents are sent to a server that extracts the attribute values and update the document database accordingly.

This paper is organized as follows: in Section 2 we show what information needs to be stored about documents. In Section 3 we present our approach in implementing the proposed solution. In Section 4 we show how the proposed solution differs from document management systems, and we discuss the advantages and disadvantages of this solution. We end up with a conclusion and a list of references.

2. STORING ELECTRONIC DOCUMENTS IN DBMS

The proposed solution to automate the queriability of electronic documents created by the employees within the organization is based on frequently collecting these documents in automated manner and storing them in a relational database management system. This database system maintains the following information (a) an electronic copy of the document so that it can be viewed or downloaded if needed, (b) information that identifies a document (e.g. sender, receiver, date and time for an e-mail received by an employee), (c) any attribute of the document that could be used to define a query or it could be the subject of querying can also be maintained in the document database (e.g., author, last access) and finally (d) the textual content of the document. The textual content of the document is stored because documents are often recalled by their textual content. The textual content is an important attribute that is needed to query about documents.

3. IMPLEMENTATION DISCUSSION

3.1 Database Design

To implement this system, we chose JavaDB (java.sun.com 2009) as relational database management system which can be easily integrated with Java applications. JavaDB supports SQL language that allows us to construct and query relational tables. The electronic files of documents can be stored in a BLOB type field. The text content can be stored as a CLOB type field. SQL provides textual based querying capabilities to query CLOB fields (using *like* construct). SQL also provides other field types that can be used to define the types of different attributes of documents (e.g. *date*, *time* and *varchar*).

For emails, we define two tables: (a) MAIL table defined by MAIL-SCHEMA=(*sender*, *receiver*, *date*, *time*, *textContent*). The fields *sender*, *receiver*, *date* and *time* identify a single received email in the employees mailbox and hence are chosen as the primary key for this table. Extracting the textual content of an email is rather simple since it is usually stored in text format, (b) ATTACHMENT table which is defined by the schema ATTACHMENT-SCHEMA = (*sender*, *receiver*, *date*, *time*, *attachment*, *textContent*). This table maintains a record for every document attached to an email (an email may have any number of attachments). ATTACHMENT has the same primary key as the MAIL table so that each attachment can be associated with its corresponding email. The field *attachment* corresponds to the binary file of the attachment which will be of BLOB type.

For electronic documents that are developed by the employee using software (e.g., Word or PDF), a different type of identifying information than emails is needed and hence a different table. We define an ELECTRONIC-DOCUMENT table that has the schema ELECTRONIC-DOCUMENT-SCHEMA = (*machine*, *path*, *fileName*, *extension*, *document*, *textContent*) where (a) *machine* field defines the machine in which the document resides. It can be a machine name or an IP address, (b) *path* is the path of the document file within the file system of the hosting machine, (c) *fileName* and *extension* define the name and the extension of the document file respectively, (d) *document* field is BLOB that stores the document file, and (e) *textContent* stores the document's textual content (if applicable). The extraction of the textual content of documents is easy for text based documents (e.g. emails). However, for other types of electronic documents, some processing is needed. For example, for Microsoft Office documents, we used a free Java API to extract the textual content (Apache POI 2002). Also we used a special API to extract textual contents of PDF documents (Apache Incubator 2008).

3.2 Collection of Documents

To maintain the document database up-to-date we need to frequently check for newly created/modified

Copy Rights

documents in the organization. New emails can be checked by software that connects with the employees' email server(s) and collects newly received emails. We used JavaMail API (java.sun.com 2009) to develop this software. This software connects with the organization's employees' email server(s), as frequently as desired (e.g., once a day). It is also used to get the needed information about emails (i.e. sender, receiver, date and time received, textual content and attachments).

To collect other electronic documents residing in the employees' personal computers, a client-server solution is needed where a client program (Java program) is installed on every machine. This program traverses the file system of the hosting machine for newly created or modified documents. We can restrict this to those files that satisfy certain extensions (e.g. .doc, .xls, .pdf) to speed this process. The client program then sends documents found along with information about the document (i.e. the name of the document, its path within the file system of the hosting machine and information identifying the machine) to the server. The client program is scheduled to run on certain time periods (e.g. once a day). This scheduling can be set by the machine's operating system (e.g. task scheduler in Windows). We implemented this client-server solution using Remote Method Invocation in Java (Grosso 2001). The server, once receives the new documents, it connects to the database, constructs corresponding records and inserts them into the database.

Note that the machine, the path within the machine's file system and the name of the document file all together identify the document. If the document was modified for any reason after being stored in the database, the document will be detected and a new version will replace the older one in the document database.

4. DISCUSSION

4.1 Related Work

Electronic document management systems and their role in information systems were introduced in (Ralph 1995). Many electronic document management systems are commercially available (KnowledgeTree 2010, OpenKM 2010, Inforouter 2010, m-Files 2010, Worldox 1998). Microsoft SharePoint (Microsoft SharePoint 2010) is a sophisticated software platform that was developed for collaboration and web publishing that also provides document management services. These systems are based on maintaining a central document repository where the users upload their electronic files. The main shortcoming of these systems is that they require employees to explicitly upload the electronic documents into the document management repository which adds extra overhead on them. So, only the documents that are explicitly uploaded by the employee are under automated management. Complex software installations, training and a change in the way with which an employee should work with documents are required. On other hand, the proposed solution is proactive in the sense that it transparently collects documents from the employee machines without overloading him/her or changing the way he/she works with documents and hence neither overhead nor special training is needed. The implementation of the proposed solution is very simple as discussed above, No expensive software installations are needed.

4.2 Advantages and Disadvantages

The main advantage of the proposed solution is that it transparently maintains all electronic documents (including emails) created or updated without overloading the employee. It does not require a complex software installation nor changing the normal way employees work with documents. It is quite easy to implement. This solution also provides automated queriability of documents whenever needed based on knowing some information about the document's attributes (e.g. author, location, textual content or modification date). This queriability can be useful in monitoring the business process activities carried out by the employees and backing up of documents against document loss due to deletion or crashes.

The main disadvantage of the proposed system is the privacy issue. The system maintains all documents (including emails) whether the employee wishes to have these documents maintained in the system or not. For emails, the system should know the user name and the password of the employee's email account. This

Copy Rights

privacy issue can be dealt with by requiring the employees to dedicate their documents and email accounts for the official use only. For emails, the organization can provide a special email accounts for the employees that are also restricted for the official non-personal use only.

5. CONCLUSION

In this paper we proposed a proactive approach to track electronic documents developed by the employees of the organization. This solution is based on frequently traversing email accounts and hard disks of the employees for newly created/modified documents, extracting their attributes (especially textual content if applicable) and then updating the document database accordingly. This system allows employees to work normally without changing the way in which they work with documents and without overloading them to manage these documents. Textual content is a very important attribute to recall documents. Extraction of the textual content of documents is possible for many types of electronic documents due to the availability of the required software APIs for this purpose. A question that needs to be answered is how to extract the textual content from documents created by different software systems. Another issue is how to control who can see which documents when querying. This requires associating the visibility of queried documents with the hierarchical structure of the organization. Also traversing the file system of the machines could be a slow process where faster traversal solutions need to be investigated.

Copy Rights

REFERENCES

Apache Incubator (2008) *Apache PDFBox - Java PDF Library*, http://incubator.apache.org/pdfbox, Date accessed 16/6/2010.

Apache POI (2002) *Java API to Access Microsoft Format Files*, http://poi.apache.org, Date accessed 16/6/2010

Inforouter (2010), Inforouter Enterprise Content and Document Management, http://www.inforouter.com, Date accessed 16/6/2010.

Grosso, W., 2001. *Java RMI*, O'Reilly, USA.

java.sun.com (2009) *Java Mail*, http://java.sun.com/products/javamail/, Date accessed 16/6/2010

java.sun.com (2009) *Java DB Easy and Advanced*, http://www.oracle.com/technetwork/java/javadb/overview/index.html, Date accessed 16/6/2010.

KnowledgeTree (2010) *Document Management Made Simple*, http://www.knowledgeTree.com, Date accessed 16/6/2010.

Microsoft SharePoint 2010 (2010) Microsoft SharePoint 2010, http://www.sharepoint.com, Date accessed 15/8/2010.

M-Files (2010) *Intuitive Document Management Software*, http://www.m-files.com, Date accessed 16/6/2010

OpenKM (2010), OpenKM Knowledge Management, http://www.openkm.com/, Date accessed 16/6/2010.

Ralph, H., Sprague, Jr., 1995. Electronic Document Management: Challenges and Opportunities for Information System Managers, *MIS Quarterly*, Vol. 19, No. 1,pp 29-49.

Worldox (1998) *Document Management System without Complexity or High Cost*, http://www.worldox.com, Date accessed 16/6/2010

ACCESSIBILITY AND USABILITY FOR PEOPLE WITH VISUAL DISABILITY

Tiago França Melo de Lima* and Janicy Aparecida Pereira Rocha**
**Universidade Federal de Ouro Preto, Ouro Preto, MG - Brasil*
***PUC-MG – Campus Guanhães, Guanhães, MG - Brasil*

ABSTRACT

The web accessibility is essential for broad citizens' access to public information and services provided by government on the Internet. However, people with disabilities often encounter barriers due to lack of conformity of websites with accessibility guidelines. Moreover, only the conformity with accessibility guidelines do not ensures usability. So, as part of an investigation over the relationship between accessibility and usability of websites for people with visual impairments, this paper presents the current stage of usability evaluation of a Brazilian e-government website for people with visual disability.

KEYWORDS

Accessibility, usability, e-government, evaluation of interfaces.

Copy Rights

1. INTRODUCTION

The access to information, a right guaranteed by the Brazilian legal system, is essential for inclusion of individuals in the Information Society and for the full exercise of citizenship. The diffusion of Information and Communication Technologies (ICTs) and evolution of the Internet allows government institutions provide information and public services on the Web. However, access and use of these by citizens is still limited because the lack of accessibility of the web sites in which are the information and services, among other factors. (Freire et al., 2008; Freire et al. 2009)

Several initiatives have been developed in order to make the Web more accessible (Brasil, 2005; WAI, 2010; W3C, 1999; W3C, 2008; W3C, 2010). However, a great number of web sites are inaccessible for many people (CGI.br, 2010; Ferreira et al, 2007; Freire et al., 2009; Goette et al., 2006; Kane et al, 2007). According the study realized by CGI.br and partners, despite the existence of a standard created by the Brazilian government to regulate the accessibility of government content published on the web, only 2% of sites in the field gov.br presented some type of conformity, in other words, 98% did not have any adherence to accessibility standards (CGI.br, 2010). Furthermore, the lack of accessibility may also generate usability issues (Ferreira et al., 2008; Melo and Baranauskas, 2006; Petrie and Kheir, 2007; Theofanos and Redish, 2003). The relationship between usability for people with disabilities and accessibility is unclear, and few data have been collected and analyzed in order to show that a higher conformance with accessibility guidelines implies on a higher usability for people with disabilities (Petrie and Kheir, 2007). Therefore, it is very important understand the relationship between usability and accessibility of websites and the necessities of people with disability (Ferreira et al., 2008; Petrie and Kheir, 2007).

The work-in-progress presented in this paper is part of an investigation of the relationship between accessibility and usability of web sites for people with visual impairments. As part of a larger work, the specific objective of this work is evaluating the usability of Brazilian e-government websites for people with visual disability. Thus, this paper presents the current stage of this evaluation.

2. ACCESSIBILITY, USABILITY AND SOCIAL INCLUSION

The ICTs, together with the World Wide Web (Web), have been increasingly used by governments to democratize access to information and public services. This process, known as electronic government (e-government), had been adopted by Brazilian government from the year 2000 (Brasil, 2010; Santos and Guimarães, 2004). However, this initiative is difficult by the digital exclusion, mainly caused by digital illiteracy, absence or limitation of access to ICT resources, and accessibility problems of websites for people with special needs. Therefore, it is a great challenge make electronic government accessible to all citizens (Brasil, 2010; Chahin et al., 2004).

According the Brazilian demographic sense of 2000, conducted by the Brazilian Institute of Geography and Statistics (IBGE), 24.5 million people (14.5% of the population) have some type of disability. Of this total, 48.1% have visual impairment partial or total (Carvalho, 2008).

The Web accessibility means provide access to Web for people with disabilities. More specifically, Web accessibility means that people with disabilities can perceive, understand, navigate, and interact with the Web, and that they can contribute to the Web (Henry, 2005).

The access of Web for people with visual impairments can be done through a screen reader software that can read the content displayed on the computer screen and through a voice synthesizer, turn it into audio output. In this context, web accessibility refers to access of the web content mediated by a screen reader (Melo and Baranauskas, 2006).

The use of Web, an environment without architectural barriers, reduces dependence on helpers for people with visual disability to access information and services (Ferreira et al., 2008). However, the accessibility becomes an essential requirement for web application's interfaces (Melo and Baranauskas, 2006).

Many efforts have been realized aiming to provide greater accessibility for web based systems. In 1999 and 2008, respectively, the Web Accessibility Initiative (WAI), a working group of the World Wide Web Consortium (W3C), published the Web Content Accessibility Guidelines, in versions 1.0 and 2.0 respectively (W3C, 1999; W3C 2008). In Brazil, in 2004, the Decree 5296 marked the first legal determination on accessibility in government websites for visually impaired. To meet the decree, was released in 2005 the Accessibility Model of Electronic Government (e-MAG), with recommendations for construction and adaptation of government contents on the Internet (Brasil, 2005; Brasil, 2010).

The conformity assessment of a website with the accessibility guidelines can be facilitated by the use of automated validators. However, this is not sufficient to identify all the accessibility problems. Other techniques must also be used to evaluate the accessibility and the usability of websites (Brasil, 2005; Preece et al., 2005; W3C 1999; W3C 2008). The evaluation can be made by specialists and / or with users' participation.

Usability is an important quality requirement that refers to how easy, efficient and enjoyable is the interaction, and also has to be considered in web based systems (Melo and Baranauskas, 2006; Nielsen, 1993; Preece et al., 2005). The ISO 9241 defines usability as "the extent to which a product [or website] can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use" (ISO, 2000).

Despite laws and policies directed to web accessibility, the number of inaccessible websites and portals is still high (CGI.br, 2010; Freire et al., 2009; Ferreira et al, 2007; Kane, 2007). Moreover, the lack of accessibility may hinders the usability of a website (Ferreira et al., 2008; Melo and Baranauskas, 2006; Petrie and Kheir, 2007; Theofanos and Redish, 2003).

3. METHODOLOGY

In order to perform the analysis of the relationship between accessibility and usability of websites for visually impaired people the following steps were defined: (i) accessibility evaluation; (ii) training of users; (iii) usability evaluation and; (iv) analysis of the evaluations' results. The DECIDE framework (Preece et al., 2005) was used in planning the evaluation. Also were used daSilva (DaSilva, 2010), Hera (Hera, 2010) and eXaminator (eXaminator, 2010) as automatic accessibility validators and screen reader DOSVOX (Dosvox, 2010) as assistive technology.

The first step aimed to identify the websites belonging to Brazilian's e-government to be used in the users training and in the usability evaluation. These websites were classified as their conformity or not to the e-MAG guidelines and will be used in later steps. The accessibility evaluation was made from the use of automatic validators and validation by human experts. As results of this step the selected websites were classified according to its conformity to the e-MAG guidelines.

The second step's goal is to train a user group, to analyze and to evaluate the process of appropriation of assistive technology. Users will be trained to access the Web using the DosVox screen reader software. The group of users will consist of disabled and non disabled people, with good experience in usage of computers and Internet, but with no experience on screen reader software. Will be used questionnaires, lectures and participative observation in laboratory. One of the questions explored in this step is the identification of main difficulties of interaction during the process of appropriation of assistive technology caused by accessibility problems.

The third step realizes the usability evaluation of a set of accessible and non accessible websites according to the e-MAG guidelines. The evaluation will be carried out from data collected by controlled observation and questionnaires. During the controlled observation the users will perform a set of tasks in laboratory and data from their interaction with the websites will be collected, being analyzed aspects like productivity, efficiency, satisfaction, ease of learning and memorizing. Also, opinion of users will be collected from questionnaires, helping to identify the main aspects of their usage experience like perception and understand about the website content, complexity of navigation, ease of use and satisfaction.

At the current stage of usability evaluation, websites to be used have already been chosen and the group of users has already been selected. The tasks that users will perform and the questionnaires were defined. Initially, evaluation will be conducted by a group of eight visually impaired users total, with experience in using computers and software screen reader DOSVOX. Subsequently, will be analyzed the users' group who have been trained in the second step. Finally, data collected will be analyzed in search of results that allow to assess how much compliance with accessibility guidelines defined by e-MAG contributes to: (i) improving the usability of Web sites, (ii) reduce users' difficulties during the process of appropriation of assistive technologies.

4. FINAL CONSIDERATIONS

The computer systems interfaces should provide a satisfactory interaction to users. It is expected that the fulfillment of accessibility and usability requirements of web based systems generates a qualitative change in information access for citizens, respecting their diversity.

Theofanos and Redish (2003) gives some reasons why the design of websites should provide accessibility and usability for people with disabilities: (i) disabilities affect many more people than we may think (750 million of people in world being more than 24 million in Brazil); (ii) it is a good business; (iii) the number of people with disabilities is likely to increase; (iv) the Web plays an important role and has significant benefits for people with disabilities; (v) improving accessibility improves usability for all users; (vi) it is morally the right thing to do. Moreover, in some countries like Brazil exist laws regulating the government content's on Web, which must be in conformity with the e-MAG accessibility guidelines. Thus, the effective inclusion of people with disabilities to Brazilian e-government websites represents the right of more than 24 million of Brazilian to have access to public information and services by digital media and exercise their citizenship (Santos and Guimarães, 2004). So, accessibility has become a very important issue to promote the inclusion of people with disabilities in the Information Society (Freire et al., 2008). However, as presented by the research of Freire et al. (2008), accessibility is still far from being actually considered in development of Web projects in Brazil. However, it is not enough to have guidelines that are based on technology to meet the needs of all users. It is also necessary to understand the users and how they work with their tools (Theofanos and Redish, 2003).

Thus, this study aims to investigate the level of usability of websites from its accessibility guidelines compliance. And a later stage to present an analysis of how much the conformity to accessibility guidelines influences the level of usability of the websites and how it interferes with in the process of appropriation of assistive technology (for people with visual disabilities).

REFERENCES

Brasil, 2005. Modelo de Acessibilidade de Governo Eletrônico. Available at: <http://www.governoeletronico.gov.br/emag>. Last access at: September/2010.

Brasil, 2010. Brazilian e-Gov. Available at: <http://www.governoeletronico.gov.br>. Last access at: September/2010.

Carvalho, C. V., 2008. Dados sobre Deficiência no Brasil. Brasília, Brazil. Available at: <http://bd.camara.gov.br/bd/bitstream/handle/bdcamara/1383/dados_deficiencia_viveiros.pdf?sequence=1>. Last access at: September/2010.

CGI.br, 2010. Dimensões e características da Web brasileira: um estudo do .gov.br. Available at: <http://www.cgi.br/publicacoes/pesquisas/govbr/>. Last access at: September/2010.

Chahin, A. et al., 2004. E-gov.br: a próxima revolução brasileira: eficiência, qualidade e democracia, o governo eletrônico no Brasil e no mundo. Prentice Hall, São Paulo, Brasil.

DaSilva, 2010. Web accessibility evaluation tool: daSilva. Available at: <http://www.dasilva.org.br>. Last access at: September/2010.

Dosvox, 2010. Projeto DOSVOX. Available at: <http://intervox.nce.ufrj.br/dosvox/>. Last access at: September/2010.

eXaminator. Web accessibility evaluation tool: eXaminator. Available at: <http://www.acesso.umic.pt/webax/>. Last access at: September/2010.

Ferreira, S. B. L. et al., 2007. Panorama of Brazilian Web accessibility. *Proceedings of the XXXI ANPAD Meeting – EnANPAD*.

Ferreira, S. B. L. et al., 2008. Tornando os requisitos de usabilidade mais aderentes às diretrizes de acessibilidade. *VIII Simpósio Brasileiro de Fatores Humanos em Sistemas Computacionais*. Porto Alegre, Brasil.

Freire, A. P. et al., 2008. A survey on the accessibility awareness of people involved in web development projects in Brazil. *Proceedings of the 2008 international cross-disciplinary conference on Web accessibility*. Beijing, China.

Freire, A. P. et al., 2009. Acessibilidade dos sítios Web dos governos estaduais brasileiros: uma análise quantitativa entre 1996 e 2007. *Revista de Administração Pública*, Vol. 43, pp. 395-414.

Goette, T. et al., 2006. An exploratory study of the accessibility of state government Web sites. *Universal Access in the Information Society*. Vol. 5, No. 1, pp. 41-50.

Henry, S.L., 2005. Introduction to Web accessibility. Available at: <http://www.w3.org/WAI/intro/accessibility.php>. Last access at: September/2010.

Hera, 2010. Web accessibility evaluation tool: Hera. Available at: <http://www.sidar.org/hera/>. Last access at: September/2010.

ISO, 2000. International Standards Organization – Standard 9241: Ergonomic requirements for office work with visual terminals.

Kane, S. K. et al., 2007. A web accessibility report card of top international university web sites. *Proceedings of the 2007 international cross-disciplinary conference on Web accessibility*. Banff, Canada.

Melo, A. M. and Baranauskas, M. C. C., 2006. Uma opção inclusiva à avaliação cooperativa de interfaces de usuários. *XXXIII Seminário Integrado de Software e Hardware*. Campo Grande, Brasil.

Nielsen, J., 1993. *Usability engineering*. Academic Press, San Diego, USA.

Petrie, H. and Kheir, O., 2007. The relationship between accessibility and usability of websites. *Proceedings of the SIGCHI conference on Human factors in computing systems*. San Jose, USA, pp. 397-406.

Preece, J. et al., 2005. *Design de interação: além da interação homem-computador*. Bookman, Porto Alegre, Brasil.

Santos, M. I. A. S. and Guimarães, A. O., 2004. Uma questão da sociedade inclusiva: a acessibilidade das pessoas portadoras de deficiência no governo eletrônico. *III Seminário Internacional Sociedade Inclusiva PUC Minas – Ações Afirmativas*. Belo Horizonte, Brasil.

Theofanos, M. F. and Redish, J. G., 2003. Bridging the gap: between accessibility and usability. *Interactions*. Vol. 10, No. 6, pp. 36-51.

W3C, 1999. Web Content Accessibility Guidelines 1.0 (WCAG 1.0). W3C Recommendation. Available at: <http://www.w3.org/TR/WCAG10/>. Last access at: September/2010.

W3C, 2008. Web Content Accessibility Guidelines 2.0 (WCAG 2.0). W3C Recommendation. Available at: <http://www.w3.org/TR/WCAG20/>. Last access at: September/2010.

W3C, 2010. World Wide Web Consortium. Available at: <http://www.w3c.org>. Last access at: September/2010.

WAI, 2010. Web Accessibility Initiative. Available at: < http://www.w3.org/WAI/>. Last access at: September/2010.

Copy Rights

AN APPROACH TO PREVENT STEMMING SIDE EFFECTS IN INFORMATION RETRIEVAL

Ahmet Arslan and Ozgur Yilmazel
Anadolu University
Computer Engineering Department
26555 Eskişehir, Turkey

ABSTRACT

Traditional usage of stemming in Information Retrieval increases recall at the expense of harming precision. However in Web Search scenario precision is more important. In this paper, we propose an approach that applies stemming with aim to improve search recall without significant loss in precision. Our proposed solution keeps original term along with its stem and gives smaller weight to stem. Implementation of our approach is done in Lucene which is an open source full text search library. Our experiments on TREC4 ad hoc task environment show that stemming hurt precision for 16 topics and we can improve precision by **26%** on these topics and **7.8%** for over all topic set.

KEYWORDS

Lucene, precision, porter, recall, stemming.

Copy Rights

1. INTRODUCTION

Stemming is the process for reducing morphological variations (plural forms, gerund forms, tense suffixes, etc) of a word to a base form. For example, the words {addicted, addicting, addiction, addictions, addictive, and addicts} can be reduced to their stem, **addict**. Many stemmers have been implemented including Porter stemmer [1].

The use of stemming in Information Retrieval is well studied and there are numerous publications on it. By its very nature, stemming increases search recall at the cost of harming precision. [2].

When stemming is employed in Information Retrieval, stem of the word is stored and original word is lost. For example, when Porter stemmer is used, all the following words {operate, operating, operates, operation, operative, operatives and operational} are saved as **oper** in the inverted index.

The example given here [3] explains what can go wrong when stemming employed. When Porter stemmer is used, the words {book, books, booking and booked} are all reduced to a same stem. Similarly the words {store, stores, storing and stored} are considered as exact same words.

- book ← {book, books, booking, booked}
- store ← {store, stores, storing, stored}

When stemming is employed, the query “book store” would return documents that contain variants of query terms and that will harm precision since some of the variants will be non-relevant for that query.

For example, the query “book store” could return following three documents: {book storing}, {booking store} and {reading a book in coffee stores}.

There are also cases where words having different meanings are reduced to the same stem. For example, Porter stemmer reduces all of the words general, generally, generation, generations, generative, generous, General and Generals to their stem, **gener** [4]. Similarly, sever, several, severally, severe, severed, severely, severing, severity and severs are all reduced to their stem, sever. Such stemming collisions hurt precision in Information Retrieval.

In this study, we propose a methodology to eliminate these side effects of stemming. In our proposed solution, we keep original word as well as its stem. So that stem becomes original word’s synonym. Also we assign different weights to word itself and its stem. We give higher boost to exact matches so they are listed in the first result page.

Bad matches described above are also retrieved in our proposed solution; however they are ranked at the end of the list. Many web users are interested in precision of first one or two pages. We used Porter stemmer in our experiments due to its popularity (effectiveness, simplicity) and wide usage. We implemented our approach in Apache Lucene[1] Java (version 3.0.2), which is an open-source full-featured text search engine library.

2. LUCENE MODIFICATION

We implemented our approach by customizing Lucene in three areas: TokenFilter, Similarity and QueryParser.

2.1 Porter Expansion Stem Filter

Lucene has already java implementation of Porter stemming algorithm. We modified PorterStemFilter so that it preserves original token in addition to produced stem, and assigns different weights to them. These weights are configurable through filter's constructor. This custom token filter emits two tokens that are at the same position, from a single word and gives different boost factors to them. These boost factors will be used in score calculation.

Boost weights are stored in Payloads[2] which makes possible to optionally store arbitrary metadata on a token by token-level. Float boost factors are encoded in byte array since payloads can store byte array only.

2.2 Payload Similarity

Lucene's default similarity mechanism ignores payload scores. To reflect token-level weights to score calculation, we override `scorePayload` method of DefaultSimilarity class so that it simply returns float value encoded in token's payload. By default this method returns 1 which is the identity element under multiplication.

2.3 Payload Query Parser

Lucene has wide range of Query types. Currently only two query types take into account payloads in score calculation. They are PayloadTermQuery and PayloadNearQuery. Lucene's default QueryParser does not utilize these queries. Therefore we override `newTermQuery` method of QueryParser class so that it returns PayloadTermQuery instead of TermQuery. This custom query parser is used to convert topics to Lucene's Query objects. PayloadNearQuery is used for proximity searches that we didn't use in our experiments. PayloadTermQuery multiplies a term's score with the value of the payload located at that term. Overall effects of payload scores for a document are calculated using a PayloadFunction. We used three different implementations that Lucene offers.

- AveragePayloadFunction calculates the final score as the average of all payload scores seen.
- MaxPayloadFunction calculates the maximum payload score seen.
- MinPayloadFunction calculates the minimum payload score seen.

3. EXPERIMENTAL RESULTS

To test our proposed solution we completed TREC-4[3] ad hoc task which consists of 567,529 documents and 49 topics (numbers 202 - 250). Note that topic number 201 is omitted in the actual competition.

[1] http://lucene.apache.org/java/docs/index.html
[2] http://wiki.apache.org/lucene-java/Payload_Planning
[3] http://trec.nist.gov/pubs/trec4/t4_proceedings.html

Copy Rights

To represent no stem option we ran experiments with StopAnalyzer that ships with the standard Lucene distribution. We added PorterStemFilter to StopAnalyzer to represent Porter stemmer. Similarly we added PorterExpansionStemFilter to represent Porter expansion stemmer.

To measure precision (P@30) we choose document level of 30. We didn't use P@5 or P@10 because they are designed for smaller test collections and are not appropriate for TREC collection. All metrics are computed by the `trec_eval`[4] package (version 8.1), based on the retrieval of 1000 documents per topic.

In our initial run we compare no stem and Porter stemming. We observed that Porter stemmer increased precision (P@30) by 4.9% for all topic set. Results of individual queries are analyzed to obtain queries which stemming decreased precision. We see that Porter stemmer hurt precision in 16 topics (number 203, 207, 208, 211, 212, 216, 222, 228, 230, 235, 237, 238, 242, 244, 245, and 247).

We focused on these 16 topics in our remaining expansion runs and compared average P@30 values. To identify best weight value and payload function combination, total thirty runs were completed. For simplicity weight of stem is fixed at 1.0 and ten different weight values (starting from 1.0 to 10.0) for original word is tested. Table I shows results of expansion runs. We obtained highest P@30 value with average payload function with weight value 2.0 for original word. In other words, giving twice importance to the original word than its stem, yield best result.

Table 1. Average P@30 Values of 16 Topics – Expansion Run

	AveragePayload	MaxPayload	MinPayload
1.0	0.4479	0.4479	0.4479
2.0	**0.4521**	0.4479	0.4395
3.0	0.4499	0.4395	0.4312
4.0	0.4500	0.4437	0.4291
5.0	0.4520	0.4437	0.4312
6.0	0.4520	0.4416	0.4312
7.0	0.4541	0.4374	0.4291
8.0	0.4541	0.4374	0.4249
9.0	0.4520	0.4354	0.4208
10.0	0.4479	0.4354	0.4208

Table II compares best representative of expansion runs with Porter stemming run and no stemming run. When compared to Porter stemming run, expansion Porter stemming technique performed 26% better for 16 topics - that Porter stemmer hurt precision - and 7.8% better for all topic set.

Table 2. Average P@30 Values – All Runs

	No Stem	Porter Stem	Expansion Stem
16 Topics	**0.4562**	0.3583	0.4521
49 Topics	0.3143	0.3299	**0.3558**

Copy Rights

Although Porter stemmer decreased precision for 16 topics in TREC-4 ad hoc retrieval task, when compared to no stemming used, our proposed expansion stemming technique successfully overcame this side effect.

REFERENCES

[1] M. Porter, 1980. An Algorithm for Suffix Stripping. Program, 14(3):130–137, 1980.

[2] Strzalkowski, T. et al, 1999. Evaluating Natural Language Processing Techniques in Information Retrieval: A TREC Perspective. In: Strzalkowski, T.(ed.) *Natural Language Information Retrieval*. Kluwer, Dordrecht.

[3] Peng, F., Ahmed, N., Li, X., and Lu, Y. 2007. Context sensitive stemming for web search. *Proceedings of the 30th Annual international ACM SIGIR Conference on Research and Development in information Retrieval*.

[4] John O'Neil, 2009. Doing Things with Words, Part Three: Stemming and Lemmatization, unpublished.

[http://www.attivio.com/blog/34-attivio-blog/333-doing-things-with-words-part-three-stemming-and-lemmatization.html]

[4] http://trec.nist.gov/trec_eval/

Copy Rights

Posters

Copy Rights

Copy Rights

IMPROVING THE RESILIENCE OF MULTIPATH TCP BY LATENCY SUPERVISION

Eng. Florin-Josef Lataretu* and Prof. Dr. Eng. Corneliu Ioan Toma**
*Alcatel-Lucent Deutschland, Thurn-und-Taxis-Strasse 10, 90411 Nürnberg, Germany
**"Politehnica" University of Timisoara, Bul. Vasile Parvan, nr. 2, 300223, Timisoara, Romania

ABSTRACT

Although today's networks operate well, they often cannot be considered resilient. A promising way to improve the network resilience and performance is initiated by the "resource pooling principle", in particular via multipath TCP. This article is providing a rough overview on resilience aspects of the multipath TCP. The main contribution is an alternative solution for improving the resilience of a multipath TCP connection by supervising the latencies on the different subflows. The proposal responds to open issues mentioned in the research agenda of the "resource pooling principle" by providing a base for traffic engineering tools which are able to anticipate how end system will shift their load.

KEYWORDS

Multipath TCP, resilience, resource pooling, latency control, traffic engineering, load balancing

Copy Rights

1. INTRODUCTION

The next-generation networks split the classical transport into different sublayer: Endpoint, Flow Regulation, Isolation, Semantic Layer (Iyengar, Ford, 2009). The Semantic Layer is in charge for application-oriented functions serving the endpoints reliability. Its main functionality is to create separate flows over multiple paths, manage end to end states cross these flows, bundle flows for shared congestion control (IETF, 2010). *Multipath TCP* (MPTCP) fits for the semantic layer functionality re-establishing "the long-lost principles of end-to-end reliability and fate sharing" (Iyengar, Ford, 2009). The fundamental shift was initiated by the "Resource pooling principle" (Wischik, et. al, 2008). The central idea is to integrate the existing mechanisms for load balancing and failure resilience into a general concept of resource pooling. The multipath TCP sets up multiple *subflows***,** which are using own window-based congestion control (IETF, 2010). Some proposals suggests that the congestion windows of the subflows should be coupled with adaptive mechanisms (Kelly, Voice, 2005), (Han, et.al, 2006). The approach in (Popa, et.al, 2006) consists of a multipath routing protocol and two congestion algorithms based on splitting the flow at the source. Predecessors of this solution have been proposed in (Hsieh, Sivakumar, 2002), (Rojviboonchai, Aida, 2004), (Dong, et.al, 2007). The congestion control in the context of the MPTCP is even increasing in complexity because the distribution on different subflows (Iyengar, Ford, 2009, Key, et.al, 207, Popa, et.al, 2006). The research agenda of the "pooling principle" (Wischik, et. al, 2008) raises following issues: predictability, late reaction, latency/buffer increase because jitter. The *packet 1+1 path protection* is an already established technology which provides a packet level protection by establishing two distinct, node disjoint connections (usually Label Switched Paths, LSPs) between the edge nodes and by treating both LSPs as working. Packets are dual fed at the ingress node on both LSPs. At the egress node only the first copy is forwarded. The decision whether to accept or to discard a packet is based on a *sliding window* which provides a range of acceptable sequence numbers. In addition a *delay window* must be considered, which reflects the number of consecutive packets the trailing LSP can fall behind the leading LSP (ITU-T G.7712, 2003). The main advantage of this protection scheme is that it recover instantaneously and transparently without requiring failure detection and protection switching mechanisms. The main disadvantage is the double bandwidth.

2. LATENCY CONTROL AND CONGESTION PREVENTION

For the end to end performance of multipath TCP it is essential to control the relative latency of the different subflows. This would require in principle comparative measurements of a kind of "MTCP ping", which would have to be generated synchronous and to travel over each of the subflows. The classical ping has the drawback of reflecting also the reverse direction. A possible option for the MPTCP is to route the subflow along a dedicated LSP, which are implemented using the well known RSVP and MPLS protocols. The existing mechanism of the MPLS 1+1 packet protection (ITU-T G.7712, 2003) can be used if it is modified as follows: 1 Extension to a larger number (N >=2) of LSPs instead of the one leading and one trailing LSP. 2 The head node replicates a regular package on a periodical interval and sends the copies with the MPLS shim header including specific sequence number on each subflow. 3 The egress node eliminates the replicas and monitors the arrival of the copies on the different subflow. This way it may get information on: relative delay between the subflows, dynamic alterations e.g. caused by emerging congestion, or by packets lost on a subflows. This information is provided as a feed back to the ingress node that may apply the appropriate reaction like redistributing the load on the subflows, creating and/or removing subflows. The proposed supervision may be performed per subflow and globally (for all subflows). The global supervision of the relative latency on the subflows is by **c**ontrol messages that are to be sent over the existing N subflows, each based on a dedicated LSP. The control message is just a certain regular TCP message including the MPLS shim header with the specific MPLS sequence number as per (ITU-T G.7712, 2003). This TCP message is replicated on every subflow containing the same value of the MPLS sequence number. The control message can be send on a periodical basis, possibly related to some requirements on the service quality and also on demand e.g. on congestion indication. They may optionally include time stamps (sending time), however from the principle they are not necessary. In general the egress node does not need any assumption of the frequency of the control message since its supervision is based on monitoring the relative delay between the one leading LSP and the following N-1 trailing LSPs. The application at the tail end usually requires a certain maximal latency, an acceptable time interval *Lreq* for getting consecutive packets. If the messages are not received inside this Lreq some counter-measurements are indicated. For the supervision a ***Current Latency Window (CLW)*** can be defined as the time difference for the control message (identified by multiple copies with the same MPLS sequence number) between the leading LSP and the last trailing LSP. The system should attempt to keep at any time the *Current Latency* below the required maximal latency, fulfilling the condition: $CLW(t) < Lreq$ *(1).* In order to detect already in advance communication problems leading to an increase of the latency affecting the end to end quality, it is indicated to define a factor *fh* which acts as a threshold value for the ratio CLW(t)/Lreq. If the threshold *fh* (high watermark) is exceeded, $CLW(t)/Lreq >= fh$ *(2),* the sender should be informed via notification, so it could take appropriate counter-measurements like reducing the load on the trailing subflow, possibly down to 0 (means withdrawing the subflow), by adding a new subflow or by redistributing to the existing subflows. This decision may be influenced by information gathered via supervision on the individual subflows (see below). The relative delays on the different subflows may provide a good basis for possible load redistribution. The value of *fh* is a matter of fine tuning taking also in consideration the size of the congestion windows. Something around 70-80% may be a good starting value. If the counter-measurements are successful, the CLW(t)/Lreq may decrease below the low watermark *fl*: $CLW(t)/Lreq < fl$ *(3)*. This should also be notified to the sender as a confirmation of the counter-measurements. Resilience may be additionally improved by the supervision of each of the trailing LSPs. For this a ***Current Delay Window*** specific for the subflow *i* (CDW_i) may be defined by monitoring the difference to the leading LSP. The system should attempt to keep at any time the latency of any LSP below the required maximal latency, fulfilling the condition: $CDW_i(t) <= CLW(t) < Lreq$ (4). For each subflow/LSP the relative variations of the CDWi should be supervised inside the high/low water marks fh_i, fl_i: $fl_i < CDW_i(t+1)/CDW_i(t) < fh_i$ (5) If these values are exceeded, then it may be an indication of congestion on the subflow. In this case, the condition should be signaled along the LSP, so that each implied node receives the information and possibly use it for some local counter-measurements. If the RSVP protocol is used for signaling the sub-flow along the LSP, then the intermediate nodes may be informed using the regular "Resv" message exchanged with the previous hop, for instance by extending the Record Route object. Besides other local adaption or correction measurements, one possible reaction of an intermediate node could be a detour around the congested, possibly failed node. This local decision may be taken based on monitoring/evaluating the final delay at the tail end against the relative delay (latency) to the neighbors of the intermediate node. Such a counter measurement may improve the situation already in advance, before affecting latency on the tail end, thus preventing end to end quality degradation. Notice that consecutive measurements of the ratio CDW_i (t+1)/CDW_i(t) results in time series which could be used for predictions of the future behavior.

3. CONCLUSION

This article proposes a solution for improved resilience of a multipath TCP connection by supervising the latencies on the different subflows. Some specific thresholds are defined to control maximal delay or jitter. If they are exceeded, the head end may initiate appropriate reactions based on the information provided by supervision. This proposal is not a substitute for the congestion control but a less 'expensive' complement acting as prevention. Based on the combined supervision information (global and per individual subflow) the escalation steps are: Local correction measurements on the intermediate node (local detour), redistribution of load on existing/new subflows as prevention, retransmissions via the congestion control The proposal answers some issues of the research agenda of the “resource pooling principle” by providing the base for traffic engineering tools, which are able to anticipate how end system will shift their load. It may provide some feed back for dimensioning the jitter buffer. The solution is evolutionary, it extends the implemented 1+1 packet protection with some slightly modifications, which are eliminating the bandwidth overhead and some extensions on RSVP signaling. For future work the thresholds defined at (2), (3), (5) could be analyzed on simulations from the perspective of their dependency with related parameter like buffer size, congestion windows, recovery times. The consequent reactions can be analyzed from the perspective of self-configuration theory and possibly extended towards an adaptive self-healing behavior of the network.

REFERENCES

A. Ford, C, Raiciu, S. Barre, J. Iyengar, 2010. Architectural Guidelines for Multipath TCP Development, http://tools.ietf.org/html/draft-ietf-mptcp-architecture-00

D. Wischik, M Handle, M.B. Braun, 2008. The Ressource pooling principle, *ACM SIGCOMM Computer Communication Review.* Vol. 38, Issue 5, pp. 47-52.

Frank Kelly, Thomas Voice, 2005. Stability of end-to-end algorithms for joint routing and rate control, A*CM SIGCOMM Computer Communication Review*, Vol. 35

H. Han, S. Shakkottai, C.V. Hollot, R. Srikant, D. Towsley, 2006. Overlay TCP for multi-path routing and congestion control. *IEEE/ACM Trans. Networking*.

Hung-Yun Hsieh , Raghupathy Sivakumar, 2002. An End-to-End Transport Layer Protocol for Striped Connections. *Proceedings of the 10th IEEE International Conference on Network Protocols*, pp. 24-33

IETF, 2010, *Charter of the Multi Path TCP Working Group,* http://www.ietf.org/dyn/wg/charter/mptcp-charter.html

ITU-T G.7712, 2003, Architecture and specification of data communication network“, Standardization Organization

Janardhan Iyengar, Bryan Ford, 2009. An Architectural Perspective on MPTCP. *Presentation of MPTCP BoF at IETF75,* Stockholm, Sweden, http://tools.ietf.org/html/draft-iyengar-ford-tng-00

K. Rojviboonchai and H. Aida 2004. An evaluation of multipath transmission control protocol (M/TCP) with robust acknowledgement schemes. IEICE Trans. Communications http://tools.ietf.org/html/draft-ietf-mptcp-architecture-00

L- Popa, C- Raiciu, I. Stoica, D. Rosenblum, 2006. Reducing Congestion Effects in Wireless Networks by Multipath Routing, P*roceedings of the 14th IEEE Internat. Conference on Network Protocols*, Santa Barbara, USA. pp 96-105

P. Key, L. Massoulié, D. Towsley 2007. Path selection and multi-path congestion control. *In Proc. IEEE Infocom.*

Y. Dong, D. Wang, N. Pissinou, and J. Wang, 2007. Multi path load balancing in transport layer. I*n Proc. 3rd EuroNGI Conference on Next Generation Internet Networks*

Copy Rights

WIN32 PE MALWARE AUTO-ANALYSIS USING KERNEL CALL-BACK MECHANISM

JooHyung Oh, ChaeTae Im and Hyuncheol Jeong
Korea Internet and Security Agency

ABSTRACT

Due to the growing number of unknown malware samples, malware auto-analysis research is now studing for analysing collected malware and making the response signature. Recently many hooking based malware behavior analysis research had proposed, but they can not analysis rootkit type malwares which directly call the kernel and avoid using the win32 api. Also, kernel-level API hooking can cause other programs to crash or perfrom unexpectedly and performance issues due to the large amount of injected code. Therefore, we present an approach based on a kernel callback mechanism to analysis lage volumes of malware sample in a short period of time. It provides a general way for drivers to request and provide notification when certain conditions are satisfied, such as creating file, eding registry entry, etc. And There is no preformace issues because proposed call-back based analysis method can monitor the behavior without injecting the hooking code.

KEYWORDS

Malware, Malware Behavior Analysis, Kernel callback.

Copy Rights

1. INTRODUCTION

Over the past several years, malware is recognized as one of the most serious threat in Internet. Malware performs malicious activities involving spam, DDoS attack, stealing personal information, and so on. Especially, 7.7 DDoS attack occurred on July 7, 2009 in Korea have caused financial damage amounting to nearly 4,500 million dollars. Also, the number of new (unique MD5) malware samples received per day by the anti-virus industry has increased dramatically over the past few years.

Hooking based analysis technique is primarily used for analyzing collected malware samples in automatic way. User-level API hooking can easily monitor the behavior of malware using injected hooking code. But it cannot monitor rootkit type malware which directly call t he kernel and avoid using the win32 API. Kernel-level API hooking uses a similar approach. However, instead of injecting code into an application, kernel-level hooking modifies kernel objects such as SSDT, IDT. In this manner, rooktit type malware is analyzed. However, this has some drawbacks. In particular, patching the kernel can cause other programs to crash or perform unexpectedly. Also, if modified kernel object is called continuously, performance issues are happen.

In this paper, we present an approach based on a kernel callback mechanism to analysis large volumes of malware sample in a short period of time. It provides a general way for drivers to request and provide notification when certain conditions are satisfied, such as creating files, editing registry entry, etc without hooking technique. This paper is structured as follows. We present background information related to malware behavior analysis in Section 2. In section 3, we describe the principles of our callback based approach and the architecture and details of system. And we conclude our paper in Section 4.

2. RELATED WORK

Until now, researches on malware behavior analysis have focused on monitoring the Win32 API calls for detecting file, process, and registry events of malware. However, more and more malware utilizes rootkit technology malware analysis has became more difficult.

Sunbelt software's CWSandbox use user-level API hooking technique that injects monitoring code around shared, for example Win32 API, which applications utilize. It changes the pointer in a process address space that point to functions and adjusts them to point to a user defined hook. However, this has some drawbacks. In particular, applications that directly call the kernel and avoid using the Win32 API cannot be monitored. Ikarus software's TTAnalyze[3] and ZeroWine[4] analyze window API call instruction set executed by QEMU. QEMU is CPU emulator which can execute instruction set of processor virtually. Therefore, TTAnalyze and ZeroWine cannot analyze behavior of the malware which has detecting virtual environment.

3. ANALYSIS APPROACH

Instead of patching the kernel with kernel level API hooking, we use kernel callback mechanism. Kernel callback is publicly supported interfaces on the kernel level that notify a process about state changes on the system. The overall architecture of our kernel callback based analysis mechanism in turn consists of 3 components, a set of kernel drivers and two user space processes as shown in Figure 1. The kernel drivers operate in kernel space and use event-based detection mechanism for monitoring the system's state changes. The kernel-driver loader, which is first user space process, loads three kernel drivers. And behavior Event Collector captures the state changes from the kernel drivers and filters the events based on process lists.

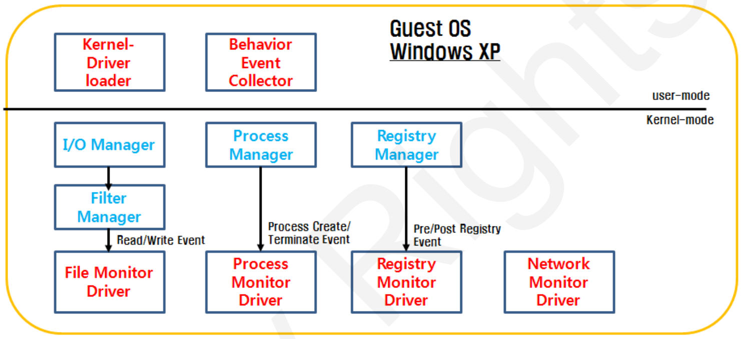


Figure 1. System architecture

Copy Rights

3.1 Process Monitor Driver

We use the Process Monitor Driver that calls the PsSetCreateProcessNotifyRoutine routine to register ProcessCallback routine which is called every time a thread performs process creation or termination.

When the Process Monitor Driver is loaded by Driver loader process, the driver calls PsSetCreateProcessNotiryRoutine to register callback function which is itself located in a Process Monitor Driver. Therefore, process related events are collected.

3.2 Registry Monitor Driver

We use the Registry Monitor Driver that calls the CmRegisterCallback routine to register a RegistryCallback routine which is called every time a thread performs an operation on the registry. When the Registry Monitor Driver is loaded by Kernel-Driver loader process, the driver calls CmRegisterCallback to register callback function which is located in a Registry Monitor Driver. Therefore, registry related events are collected.

3.3 File Monitor Driver

File Monitor Driver works differently than other drivers. Instead of registering callback function, the file monitor driver registers itself as minifilter driver that sits between the I/O manager of the Windows kernel and the base file system.

The Filter Manager is a file system filter driver provided by Microsoft that simplifies the development of third-party filter drivers and solves many of the problems with the existing legacy filter driver model, such as the ability to control load order through an assigned altitude. A filter driver developed to the Filter Manager model is called a minifilter. Therefore, we develop the file monitor driver as minifilter driver. After it has been loaded, the minifilter driver proceeds to register with the filter manager on what events it wants to listen to. Therefore, file related events are collected.

3.4 Network Monitor Driver

We develop Network Monitor Driver as NDIS IM driver to monitor network traffic. Network Monitor Driver implements two types of interfaces; protocol interface and the miniport interface. The miniport driver communicates with the miniport interface and the protocol driver communicates with the protocol interface; both of them reside in the network monitor driver. Therefore all network traffic that is being accepted by NIC card can be controlled and monitored by our network monitor driver.

3.5 Kernel Driver Loader and Behavior Event Collector

Kernel Driver Loader and Behavior Event Collector is an application that resides in user space. Kernel Driver Loader is responsible to load four kernel drivers. And Behavior Event Collector is responsible to collect behavior events received by the drivers. Once Kernel Driver Loader loads kernel driver, Behavior Event Collector creates a shared memory space and passes its address to the kernel drivers.

When behavior event is collected, kernel driver copy the event data into the shared memory space. Behavior Event Collector periodically access and check that newly event is exist. If there is new behavior in memory spacer, behavior event collector reads the event. Using thie method, kernel driver and user space process communicate each other.

4. CONCLUSION

Due to the growing number of unknown malware samples, most of anti-virus industry builds a malware auto-analysis system that can determine which samples are really malicious. Hooking based analysis technique is primarily used for analyzing collected malware samples in automatic way. However, hook based auto-analysis cannot analysis rootkit type malwares which directly call the kernel and avoid using the win32 API and cause other programs to crash or perform unexpectedly and performance issues due to the large amount of injected code.

In this paper, we present an approach based on a kernel callback mechanism to analysis large volumes of malware sample in a short period of time. It provides an advantage of kernel-level hooking without performance issues and cannot cause other programs to crash or perform unexpectedly.

Copy Rights

ACKNOWLEDGEMENT

This work was supported by the IT R&D program of MKE/KEIT. [10035427, The Development of Automatic Analysis and Malicous Site Detection Technology Against Malware]

REFERENCES

Artem D. Et al, 2008, Ether: Malware Analysis via Hardware Virtualization Extensions, *Proceedings of the 11th Information Security Conference*, Alexandria, USA, pp. 191-203

Clemens K. Et al, 2009, Effective and Efficient Malware Detection at the End Host. *Proceedings of USENIX Securitty,* Montreal, Canada, Bologna, Italy, pp. 29-36

CWSandbox, http://www.cwsandbox.org

A METHODOLOGY FOR COLABORATIVE AND COMPONENT BASED SOFTWARE DEVELOPMENT

Jonathan Bar-Magen Numhauser, José María Gutiérrez Martínez, Luis De Marcos
and Jose Antonio Gutierrez De Mesa
Computer Science Department
University of Alcalá
Alcalá de Henares, Madrid, Spain

ABSTRACT

This paper intends to propose a development methodology for Community and Component Based Software Development. Following the theoretical analysis and practical application that took place during the last year, we introduce this methodology and show the use of collaborative tools in the development process. Consequently a dedicated explanation of the group work is explained in details, which as a result allowed us to obtain a number of working roles that take part of the methodology.

We dedicated the last section of the paper to connect the collaborative tools with the methodology obtained. The impact of these tools on the methodology reflects the core idea on which the methodology thrives, from one side the collaboration work and in the other the component sharing activities between the global working groups.

KEYWORDS

Collaborative Communities, Component Methodology, Software Engineering, Google Wave, Virtual Communities, Component development.

Copy Rights

1. INTRODUCTION

This paper introduces the use of the component community based software development methodology. Component based development has been a part of the software engineering field for a long time and lately it has been passing through a transformation that affects the development process.

In this work we studied the working tendencies in a number of development frameworks, and introduced some improvements to the already established Agile Methodologies with the final objective of obtaining a methodology that encapsulates the components base development process as well as the collaborative nature of the working process. Once we came up with the methodology, we added a technical support by introducing the possible tools that can be used to improve the collaborative work.

As it can be seen in most recent surveys[1], Java is the main SDK (Software Development Kit) for software development thus giving the Object Oriented development processes a center role in most of the development methodologies. Nevertheless Java also intended to introduce us to a more abstract level of development, the Component Based Development [1].

In the last few years a development Framework for CMS [2] (Content Management Systems) web applications allowed the Component Methodologies to acquire a more privileged position. As the Framework became more popular among the developers, this new development process started to raise new questions on the general overall process. One of this Frameworks is *Joomla!*.

Joomla! is a CMS based on the PHP programming language, that brought to the surface the use of Component based process development introducing to us a new point of view for the use of this method. This CMS Frameworks thrives on its community that feeds the component market and that eventually allow local developers to acquire components for their Web Applications, install them and use them without the need for

[1] http://langpop.com/

further modification at a programming level, which eventually implies that a complete Web Application can be developed without the need of writing any line of code. [2]

In this paper we discuss the above stated points, and analyze to a more detailed level the effect of Community and Component based development on the software engineering with the objective of introducing the new elaborated working methodology obtained from this study.

1.1 The Problem

The main obstacle that rises from all this is precisely the complexity of the work process and the lack of an established methodology that can be adapted to these new activities and cover the needs of a standardized work process.

In the following sections we explain a solution to this problem by offering a proposal for a new organizational structure and finally achieve an evolved methodology that covers the Community and Component Based Software Development Methodology.

2. COMMUNITY AND COMPONENTS BASED DEVELOPMENT METHODOLOGY

Copy Rights

To establish our proposal of a work process for a Components and Community based project, we see fit the need first to divide the development work in two macro levels, the **Global development process and the Local development process.**

From now on we will consider the existence of these working areas mainly because the development of Components and Community based applications are translated to a local development and implementation of the projects, and in the other level the distribution and feedback to the global community by supporting the global Framework project.

The first level is the Local components based methodology. Any project has to be addressed by a group of developers that are either physically present or share the same work space, or collaborate through collaborative tools on the internet. In both cases it has to be considered as a decentralized work process, as the work is taken forward by the local developer.

The work process on the Local level will be strongly connected to the Global level. When a development process takes place on the local level, the developers are constantly in contact with the community and the global framework project, acquiring documentation, examples, and components that may optimize the overall development process.

Finally each local development team will form part as a member of the global development team, and as was stated before will constantly offer feedback to the community and thus improving the quality of the Framework project, and complying with the decentralization ideal.

2.1 Local Development Methodologies, SCRUM and XP

During the *Joomla!* Master Class event, that took place in the month of February 2010 in the University of Alcala, the main speaker and one of *Joomla!* founders, Alex Kempkens together with our support used an extreme programming methodology to represent what can be described as a Local level development. The chosen methodology was the SCRUM methodology [3]. To this methodology we introduced a number of new profiles that are the results of our study on real practical situations.

These profiles are to be considered as an essential part of the proposed methodology being idealized in this paper. The profiles that will interact in a Local development team are:

- **Project manager**: Will work at all levels, on the back and front end of the CMS.
- **Components analyst**: Works on the back and front end, with a special dedication to components analysis, design and component development.
- **Programmer**: Works on the back and front end, with a special dedication to components development.

- **Graphics designer**: His responsibilities will be especially in the front end, even though he will have some work on the back end in case of a component development. He will be in charge of templates development.
- **Content manager**: Both Front and back end. Will be in charge of the maintenance, and updating the project content. It's a fundamental profile of this methodology.
- **Final user**: Will work together with the Content Manager. This profile will be in charge of executing validation tests and analyzing the accessibility of the project.

2.2 Global Component based Software Development

Each local team will have to interact with the global community, and search for new components and collaborative help to reduce the effort on the local work [4].

The first profile, fundamental for the proper function of this method is the **Global Project Manager (GPM)**. The GPM's main responsibilities will be first to **keep up to date the local projects**, in which he is implicated, and secondly to keep himself updated on the global Framework components repository as new components may be developed at a local level, in other global points as well, and published at the global level.

The impact of the GPM is not only on the local projects inside his developing group, but also on the global community, mainly because **this kind of Frameworks are a community based technology on which all local projects feed from, and to which future components are intended to be distributed through.** One final attribute that the GPM will have is the ability to open closed projects. Many components created for old and deprecated projects may find new use in new projects. This fact obligates the GPM to have access to high level documentation as well as a global overview of the projects situation.

The GPM role is intended to replace the standard Project Manager seen in the local level.

The second profile is the **International Communication Manager**, ICM. The ICM's main role is to function as a **pipe line between the global community and each of the members of the local development group**. Each member of the local group may need to communicate with other members of other local groups of the same expertise. The ICM will function as a mediator and talk between members of the same level with different local groups, to improve the working flow. In other words, most of the communication between a graphic designer of local group A and a graphic designer of local group B will take place through the ICM as a mediator.

3. SUPPORT TOOLS FOR THE COMMUNITY DEVELOPMENT ACTIVITY

To ensure the work flow on the local and global levels there will be a need to make use of collaborative tools. Some of these tools are the web pages that offer various methods of communication, Forums, FAQ, and statistics. Yet we see fit the need to mention a more recent published tool, Google Wave (Wave in a Box), Novell Pulse, and Eureka Streams created by Lockheed Martin. These two tools opened the scenario to a new scene in Collaborative Software Development. Google Wave (Wave in a Box) offers a communication platform for all activities, but its impact on the professional field of work has been much more significant than in any other field.

With better collaborative tools, the local and global teams formed by the profiles mentioned in the last chapter will be able to acquire a better set of components, in a more agile way and in a less amount of time, resulting in a lower cost in the development process [5].

4. CONCLUSIONS

The objective of this paper was to offer a possible solution to the new situation created by the tendency to use the Collaborative strength of a focalized community on a certain technological development Framework, and to refresh and introduce the use of the Component based development in this process.

Copy Rights

As a result we obtained a proposal for a development methodology destined to the new generation Software Development, based on the support of an active community and the use of component based software development.

The manner in which this methodology can be applied to both of this technologies may vary, but the fundamental structure will remain, the use of a well-established community to offer a constant Feedback to the Local development group through the intervention of the specified profiles, as may be the GPM and the ICM, and the creation of software based on the use of recycled components, **thus reducing significantly the coding and creation from scratch of new components**.

Finally the collaborative tools will have a great effect on the functionality of this methodology by inserting a new workflow, offering a better support for the team work **and improving the communication between the Local groups through the Global level**.

REFERENCES

[1]Matena V., Stearns B., Damichiel L, 2003, *Applying Enterprise JavaBeans: Component-Based Development for the J2EE Platform, 2 edition***,** Pearson Education.

[2] LeBlanc J, 2008, *Learning Joomla 1.5 Extension Development*, Packt Publishing, Birmingham UK

[3] Murphy C., 2004, Adaptive Project Management Using Scrum, *Methods & Tools Global knowledge source for software development professionals,* Volume 12 - Number 4, pp. 10-23.

[4] William C. Wake, 2002, Extreme Programming as Nested Conversations, *Methods & Tools Global knowledge source for software development professionals,* Volume 10 - Number 4, pp. 2-13.

[5] Fross M. Wake, 2003, How to Select a QA Collaboration Tool, *Methods & Tools Global knowledge source for software development professionals,* Volume 11 - Number 1, pp. 26-31.

Copy Rights

DIGITAL SLIDESHOW PERFORMED LIVE USING THE "MOTORWAY" APPLICATION

PhD student Cristian Ţecu, PhD student Adrian Popescu and Prof.dr.eng.Radu Vasiu
"Politehnica" University of Timisoara, Romania
Pta. Victoriei, nr. 2, 300006 Timisoara, Romania

ABSTRACT

The development of the digital technology has over passed the old way of presenting slides, also known as Diaporama. The emergence of the specific applications allowed the passage to the digital slideshow, where every presentation can be precisely programmed. The only missing part is the lack of a real time presentation, a live one. The article tries to offer a solution for presenting a slideshow using offline and/or online images, with real time transitions, whose duration is totally controllable using the keyboard.

KEYWORDS

AV, Diaporama, Slideshow, Transitions, Internet Explorer.

Copy Rights

1. MOTIVATION

This development is dedicated to an expansive domain, the digital slide show. The digital (r)evolution brought both the photography and the information exchange into the mainstream, which led to a major expansion of the audio-visual presentations. PowerPoint became one of the main tools for this kind of presentation, but the slide show professionals are using specialized, helpful applications [Russell, Amernic, 2006]. ProShow Gold, Pictures to EXE and Wings Platinum are leading the specialized applications market, which led to a rebirth of the European slide show movement [Ondina, 2008].

The downside of these applications is, in our opinion, that they do not offer an interactive, live control of the presentation; they do not facilitate the real-time control of the image transitions, as it could have been done easily with the analogical devices. This lack, uncovered by the digital applications, inspired and motivated us to find a practical solution for simulating the analogical devices of presenting a slide show, but with digital means.

Using several platforms, we have conceived and implemented three applications that allow real-time full control of a slide show. The first two of them have been already presented at the IADIS conferences" [Cristian Ţecu et al., 2008, 2010]. Their limitations motivated us to use a different approach. The result, an application named „Motorway", is the closest to the starting concept, and the evaluators' results confirmed that.

2. THE "MOTORWAY" APPLICATION

The application has been conceived keeping in mind some compulsions and constraints.

1) Portability – the application should be used on most of the computers' configuration, without installing additional applications.

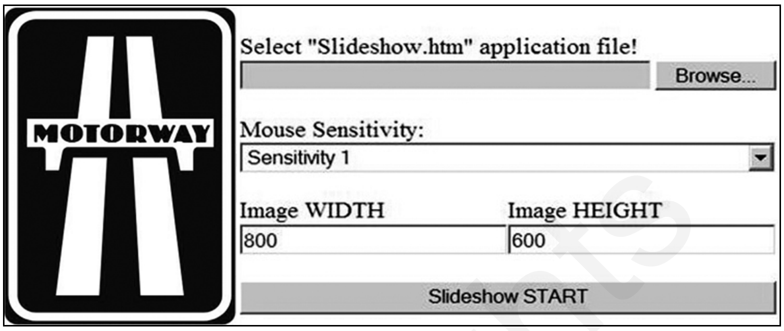
2) The „cross-dissolve" transition processing should be done without a mathematic calculation which lasts, in some situations, more than the required transition length. Internet Explorer has already implemented the mechanism that controls the „cross-dissolve" transition function, and it can solve this requirement; it is supposed to be installed on almost any Windows-using PC [Shelly, Freund, 2009]. Firefox was a second

option, but there are different standards, compared with Internet Explorer, thus they are not 100% compatible.

3) The file management was another constraint. The user is free to copy the application wherever he wants, and by defining its start, it can begin to work. Choosing the “slideshow” file, we define its partition and root by default; the start has to be defined manually. These were the limitations and constraints that have arisen in implementing the application; ultimately, this is just a webpage.

For developing the application, we have used JavaScript (it makes the web pages more interactive), HTML (the base of any web-page) and CSS (tags used for web-pages formatting).

Internet Explorer has also an internal mechanism that runs audio files, i.e. Windows Media Player. The advantage is that it is not necessary to start in parallel another audio application. It can play audio files compressed in .mp3 format or uncompressed (.wav).



Select "Slideshow.htm" application file!

Mouse Sensitivity: Sensitivity 1

Image WIDTH 800

Image HEIGHT 600

Figure 1. "Motorway" application browser window

Using the keyboard and/or the mouse, the slide show can be fully and interactively played and controlled. The keyboard controls command the start / pause of the music, and steps to previous / next song. The numeric pad controls the transition time, each key is assigned the equivalent time in seconds, from 0 to 9 seconds. The keypad allows to reverse the alphabetical order of the slides, or to pause the transition.

The mouse accuracy can also be controlled, by adjusting the operational length, in three steps. The keyboard control simulates the analogical commands, performed by the Kodak S-AV Programmable Dissolve Control. The mouse control simulates also the analogical commands, performed by the Simda or Imatronic systems. Thus, the application solves the key-problem, to control a digital slide show in an interactive way, using the keyboard and/or the mouse.

For opening the application, in the setting window, the “slideshow” file has to be select. Display resolution and mouse sensitivity have to be chosen. The application identifies the loaded images and the soundtrack, counts and orders them alphabetically. We have to choose the display resolution, regardless of the actual image size, because the application does not perform a resize of the images. If the display resolution is smaller than original image size, only a part of the image will be visible on the screen, its center, whose size will be equal to the size of display resolution. If the display resolution is larger or equal to the original size of the image, it will fit entirely on screen.

Copy Rights

2.1 Testing and Evaluating the “Motorway” Application

When assessing our application, one should consider that it is an interdisciplinary initiative - IT, education, and media. We have done the application’s evaluation using the Zef platform [Zef Solutions, 2009].

Motorway application’s testing was performed by sixteen reviewers with different activities, divided into two groups; one produced artistic audio-visual presentation, while the other’s involvement was on educational presentations.

The conclusions reached by the evaluators following the tests were collected through a questionnaire implemented using the Zef platform. Based on analysis of their replies, we could draw the following conclusions. Both the two-dimensional graphics used to test the ease of use and usefulness, and the graphics that show the quality and speed of response, collected all the responses in the best quadrant (above average values), indicating that the application is usable and useful, that it provides a quality solution in terms of an interactive audio-visual presentation.

As a general conclusion, we can state that the application’s assessment and testing reveals its potential, both in an academic environment as well as in the artistic, educational, commercial, advertising environment. It can be improved and developed, which should lead to an increase of its attractiveness and importance.

The reliability of the implemented methods was evaluated by testing them from several perspectives. We found an increased enthusiasm of the evaluators. We found encouraging that although they were divided into two groups, both of the responses were positive.

During our research [Ţecu, 2010], we have brought many personal contributions to the development area. The most important are described in the next paragraph.

3. CONCLUSIONS AND CONTRIBUTIONS

Studying the analogical and digital slide show, we have concluded that although the digital version replaced the analogical one, the possibility to interact with the presentation has been reduced [Ţecu, 2010]. The slideshow producers prefer ProShow Gold, Wings Platinum and Pictures to EXE. We made sure that no application has used our idea, described in this paper. We consider the idea of a digital device to simulate the control of a slideshow-type presentations, emulating similar analog devices, belongs to us.

This idea popped up while using PowerPoint; we noticed the lack of control for the transition length. Studying dedicated slide show applications, we found that they also do not fill this gap. Using the existing keyboard instead of external control devices simplifies everything. Of course, all applications running on a computer are controlled by the keyboard. However, we consider the idea of using the numeric keypad as a temporal equivalent of the digital transition controller, is ours. Because the analogue remote control was the instrument that allowed the real fine control of the transition, we were thinking to build a similar device, but digital. However, such a specialized device would have been complicated, and the application would not have been accessible to other users. The idea of using the mouse as a slideshow controller makes things much easier, and thus the portability goal is achieved.

Since our previous works, “Digital Diaporama” and "PhotoSlide Toolbar” [Cristian Ţecu et al., 2008, 2010] had shortcomings related to the installation or to the transition quality, we designed the actual application, which gathers our experience gained during the digital slideshow study. We have conceived and implemented an application, which is a viable and universal solution to present a slideshow in interactive mode. The only requirement imposed is the existence of the Internet Explorer browser installed on the PC. "Motorway" represented the final stage of study, ready to be offered to those interested. It can be downloaded from http://cristi.cm.upt.ro.

Copyright

REFERENCES

Joaquín Perea, 2001, "Audiovisuales basados en la diapositiva: El diaporama y la multivisión," *Universo Fotográfico*, vol. Nr 4, pp. 129-157.

Craig Russell, Joel Amernic (2006) "PowerPoint Presentation Technology and the Dynamics of Teaching ", Innovative Higher Education (Volume 31, Number 3 October, 2006) Pag:147-160.

Patricia Ondina, 2008, *Diaporama numerique*: Editions Générales First, 2008.

Cristian Ţecu et al., 2008, "Contributions to the use of the new computer technologies in the digital slideshows," in *IADIS International Conference Computer Graphics and Visualization 2008*, Amsterdam, pp. 311-314.

Gary B. Shelly, Steven M. Freund (2009) "Windows Internet Explorer 8: Introductory Concepts and Techniques", Course Technology, Shelly Cashman, ISBN: 9780324781670

Cristian Ţecu et al., 2010, "PhotoSlide Toolbar: Using the Internet browser for managing real-time digital slideshow," *In IADIS International Conference e-Society 2010*, Porto, pp. 503 - 506.

Zef Solutions 2009, [Online], web page: http://www.zefsolutions.com/en/solutions.html

Cristian Ţecu, 2010, „Contribuţii la utilizarea noilor tehnologii informaţionale în diaporama digitală”, Editura Politehnica, Timişoara. ISBN: 978-606-554-156-6

Copy Rights

AUTHOR INDEX

Copy Rights

Copy Rights