Accessibility of E-Cloud: An Algorithm for Implementing Interface Specification for Visually Impaired

Mohammed Fakrudeen, Sufian Yousef, Abdelrahman H. Hussein

Abstract — **This paper describes the architecture of cloud service and technologies for implementing accessible services for visually impaired (VI) users. The cloud computing emerges as a natural platform for learning. The Educational Cloud (E-Cloud) computing provide terminals with powerful e-learning resources. However, these e-learning resources are not accessible for VI users. The paper discusses the cloud service architecture and key algorithm about implementing user specification in interface design of various devices. The proposed accessibility service is available, scalable and adaptable to VI users and different type of device. As far as is known, this is a novel suggestion.**

Index Terms—**Cloud Service, Cloud Computing, M-learning, User Specification, Accessibility Service, and visually impaired.**

I. Introduction

As technology changes rapidly, Cloud computing provides a comprehensive platform for users, without bothering about the physical infrastructure and technical implementation such as platform as a service, software as a service and web service. Cloud computing is delivered on-time, regardless of location and time through network [11]. With minimum service provider interaction, hardware and software of cloud computing can be accessed without large capital outlays. Thus, the cloud computing encourages students to more proactive approach to learning [10] as it enables interaction with educational resources through their handheld devices such as smart phone, tablet and so forth.

However, the cloud computing is not utilized to provide accessibility services for disabled person[1]. When the visually impaired (VI) user access the application through the smart phone device, the VI user specification has to be set to adjust the interface design such as zooming, increasing font size and so forth. This has to be repeated whenever the VI user access the same application in different devices such as tablet, smart phone and so forth.

Manuscript received March 14, 2014; revised April 8, 2014.

Mohammed Fakrudeen is with the Anglia Ruskin University, Chelmsford, UK and also with Department of Computer Science and Software Engineering, College of Computer Science and Engineering, University of Ha’il, PO Box 2440, Ha’il, Saudi Arabia (e-mail: m.fakrudeen@uoh.edu).

Sufian Yousef is with the Department of Engineering and the Built environment, Anglia Ruskin University, Chelmsford, UK (e-mail: sufian.yousef@anglia.ac.uk)

A. H. Hussein is with the Department of Computer Science and Software Engineering, College of Computer Science and Engineering, University of Ha’il, PO Box 2440, Ha’il, Saudi Arabia (e-mail: ar.hussein@uoh.edu.sa).

The purpose of our research is to derive the accessibility service through cloud computing, to enable adaptability of interface design setting according to VI user needs ,when the application is opened in any device. The objective of the present study is to explore educational cloud service (E-cloth) and analyze the VI user interface design specification to be implemented in the touchscreen environment and their strategies in dealing with the device.

II. Educational Cloud and M-Learning

According to cloud computing, m-learning can considered as education via app-as-service. It requires application to be downloaded from mobile marker or browse the web through mobile devices. It reduces the burden of learners to get latest update without any charges. And also reduces the maintenance and support of educational institutions to the vendor.

Masud and Huang[2] summarize the consequences and implication of educational cloud as: 1) accessed via web or through application; 2) subscription may be free or paid based on usage; 3) additional software not required; 4) All learner data held on cloud server; and 5) Cloud server may support many educational institutions.

The other advantage of e-cloud is: backing up data is not required[3], crash recovery is unwarranted[4], can be accessed with various devices such as mobile, laptop and iPad , accessed from various places such as school, library and home[5], impossible for learners to identify data(exam results, question papers) storage[4] and, monitoring of data become an easy process[6].

Tools used in e-cloud are changing rapidly and in development. The traditional tools can be installed in school owned devices and limited availability of web based devices such as chrome books or on mobile devices. These tools are much advantageous and increase the learning and reading efficiency of the students. The review of literature reveals that the tools available in e-cloud can be categorized as text-to-speech, speech-to-text, visual thinking, demonstration and supplementary tools[7].

1. *Text-to-speech tools*: Read & Write for Google, Speak It, Chrome Speak, iSpeech, Kaizena and App Write Cloud.
2. *Speech-to-Text tools*: Voice Note, Dictanote and Dragon Dictation
3. *Visual Thinking tools*: Padlet, Mindmeister, Lucid Chart and Popplet.
4. *Demonstration tools: Google docs, Spreadsheet and Slides, YouTube, Vocaroo, Go Animate and Minecraft.*

5. *Supplementary tools*: Readability, Google Translate, Google dictionary, Google Thesaurus, Evernote and Summarize This.

Although the e-cloud is more beneficial for institution, educators and learners, it has some limitation: 1) the educational institution has to pay for site licensing with other technical support for the individual software package[8]; 2) mostly resources are under utilized during the night and semester break; 3) System need to be configured if the workload is high and it is expensive; 4) needs internet access to utilize e-cloud; and 5) the accessibility for a disabled person is not cared of.

III. E-CLOUD LAYERS

The E-cloud services for m-learning include cloud architecture and service method. E-cloud as service-oriented architecture can be classified into hierarchical model based on type of service provided for learners.

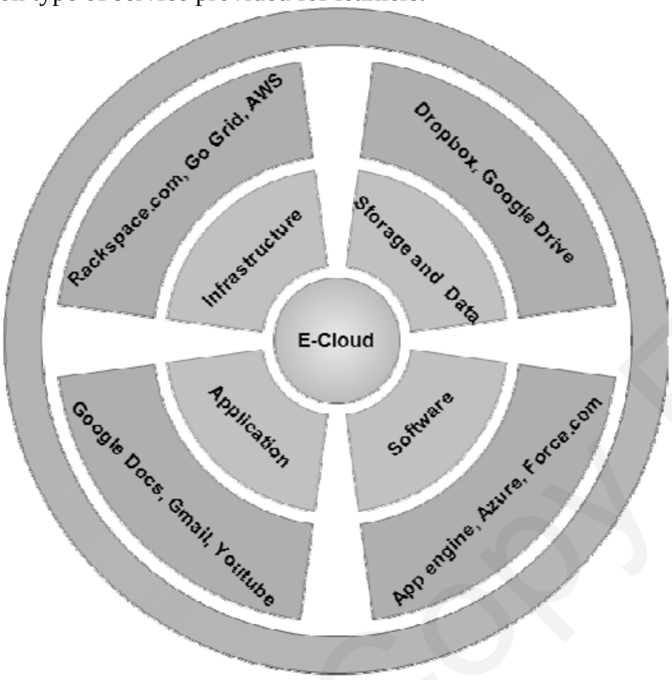

Fig 1. E-cloud Layers

An E - cloud service of hierarchical model is represented in Figure 1. It includes infrastructure (hardware), storage, data, software and application. According to e-cloud service life cycle from origin to the application, the user can be divided into end customer (learner and educator), developer (educator, software developer) and system admin.

To provide learners with convenient service at low cost is the main aim of cloud service. The learner or educator can access the cloud services by service proxy. It should be ensured to have unrestricted, yet safe access for the students.

Cloud models do not restrict to any platform and provide a multitude of devices. Thus, learners do not need to choose their storage and hardware. Data storage tools such as Drop box and Google drive allows the students to store their data from any device. The students have the options of choosing any hardware platform such as the Apple iPad running on the IOS operating system, laptop running Windows to access their stored data by means of internet.

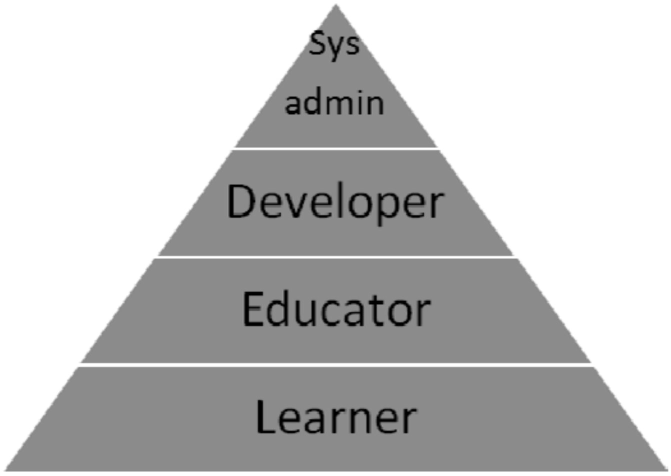

Fig 2. E-Cloud users

The application and software tools can be accessed through a web browser through learning, collaboration and sharing methods and learners can choose to use over the cloud. Some tools are available in offline. However, the internet is very much needed for delivery, sharing, and collaborative aspects. Many mobile devices can access cloud based resources.

IV. E-CLOUD FOR BLIND AND VISUALLY IMPAIRED LEARNERS

The blind or visually impaired (VI) learners accessing cloud services using touch screen devices have more usability problems due to[9]: 1) large information is displayed in small screen size; 2) gestures to access the information varies with the different technologies; 3) limited availability of free external tools to access the information, and 4) unawareness about the built-in accessibility features of the devices.

Thus, when the blind or VI learners access the e-cloud, the e-cloud should provide the interface according to user specification. The VI user differs with the sighted user in interface specification such as background color, foreground color, font and content size. These specifications depend on a device used by the VI learners.

Currently, the cloud has not implemented an accessible service for the blind user. Hence, it will be challenging for VI users to access e-cloud. As a result, it is imperative for blind or VI user to provide accessible services.

V. USER SPECIFICATION MODEL

The developing accessible application has become increasingly more complex, less productive and more costly. The adaptability according to user need is very important considering the environment and user requirements where the application is generated.

A user specification (US) model is created in our system contains all information necessary for the accessibility of UI and will also serve as the basis for future template for other services.

A US model contains two types of elements: 1) administrative information, required to understand the user's data; and 2) specification information, which are related to UI design that will help VI to interact with UI easily. Other information may be added to the model depending on the requirements. An XML format is used to store all the information.

ISBN: 978-988-19252-7-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

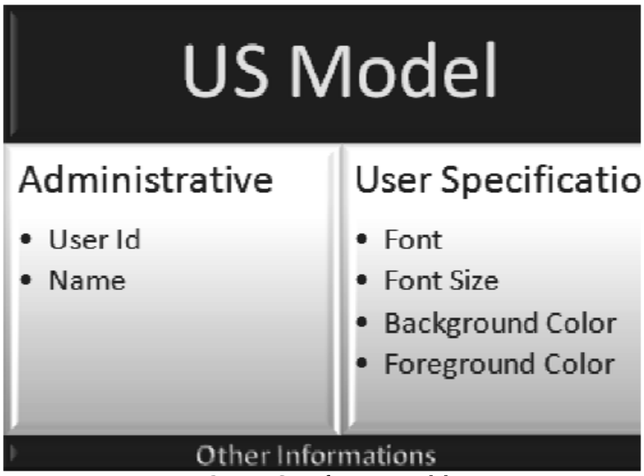

Fig 3. User Specification Model

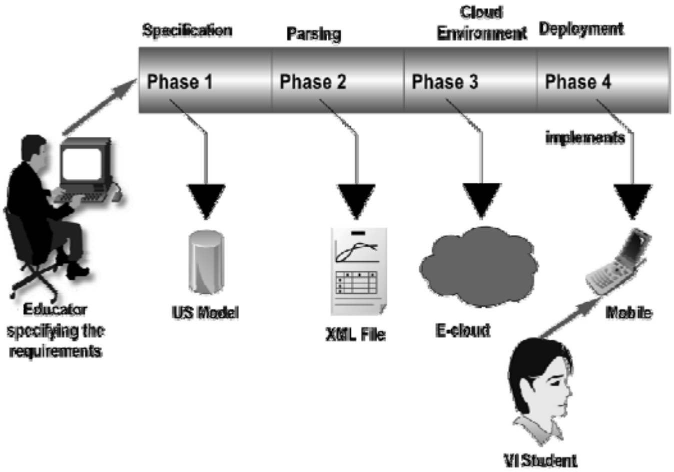

Fig 4. Specification Scenario.

Some advantage of this model:

1. A US model contains all information necessary for accessing user interface by VI users.
2. Modification of the specification is allowed in single artifacts and then other activities are updated accordingly.
3. Other formats such as CSV can be used to generate the specification information to help cloud computing provides to user the format which they preferred.

VI. US SCENARIO IN CLOUD COMPUTING

The process of using user specification in the E-cloud computing consists of four phases. The VI students or educators interact in Phase 1 to specify the user requirements and in Phase 4 to implement the specification in e-learning application in user device. The Phase 3 and Phase 4 are automatic or manual process based on requirements.

The purpose of each phase is discussed below:

Phase 1: The educator access the portal and consults the available specification items with the student to verify whether the specification is suitable for VI students to access the UI. It will be stored in the database as US model.

Phase 2: After creating the US model, all the specifications of VI students and their data, will be generated in an XML format.

Phase 3: The developed XML format is deployed in E-cloud computing, which needs the intervention of the educator or cloud service provider.

Phase 4: Whenever VI students open the e-learning application, updated XML file will be downloaded automatically to the device. The Student has to input the information such as user id for the identification first time, the e-learning application search for the user information in the downloaded file and retrieve the specification details. It is then implemented in the UI to be ready for accessing by blind students.

VII. ACCESSIBILITY SERVICE ALGORITHM

Pages are typically read one word at a time from top left corner to the bottom right corner. Although the page is designed to communicate with the reader, it is difficult for VI reader to control an overall picture of the page and the message at the same time.

There are lots of chances to lose navigation control due to color, font and contextual information. Non familiarity with the pages, look like it has not been visited before. Thus, it is imperative to set user specification according to the device the user used. The implementation of user specification for accessibility is two-step algorithm: user specification generation and XML parsing algorithm.

UI Specification Generation Algorithm. The user specification procedure can be illustrated as follows. First, the educator or VI students set the specification such as userid, background color, foreground color, font, font size and other implementation such as device specification if required. Second, these specifications are stored in the database with the userid as record identifier. The first and second steps are combined in the Phase1. Third, the XML is generated using the user specification database. This is attributed to Phase 2. The algorithm for XML generation is represented in Figure 5.

The generated XML file is uploaded to the E-cloud either manually or automatically which is application specific.

XML Parsing Algorithm. The XML parsing procedure can be illustrated as follows. First, when a user opens the application in user device, accessible service request is sent to the E-Cloud. Second, the E-Cloud sends the XML file into application. Third, the application has XML Parsing procedure which pulls XML data using HTML script. The algorithm for extracting XML data is represented in the Figure 6.

Finally, the extracted data are set to interface design.

```
SetSpecification()
[
    // procedure to set the specification
    // userid is the identification of user
    //bgcolor is the background color required for VI student
    //fgcolor is the foreground required for VI student
    // font is font name such as Arial and so on
    //fontsize is the size of the font prefered by the VI student
        XMLWriter.StartDocument(true)
                // XMLWriter is the instance of XMLTextWriter
            userid as String, bgcolor as String,fgcolor as String,
            font as String,fontsize as integer
                // variable declaration
            Connection.open() // code for connecting to the database
                do until not EOF  // loop for retrievin all records
                // retrive UI specification records and
                // place in the appropriate variable
                        do while readRecord // loop for current record
                        CreateNode(userid,bgcolor,fgcolr,font,fontsize,XMLWriter
                            //calling CreateNode procedure
                        loop
                loop
            Connection.Close()  //close database connection
        XMLWriter.EndDocument(true) //close XMLTextWriting
]
CreateNode(userid as String, bgcolor as String,fgcolor as String,font as String,
            fontsize as integer,XMLWriter as XMLTextWriter)
[
    // procedure to create nodes in XML format for given parameter
    XMLWriter.WriteStartElement("user") // opening tag
        XMLWriter.WriteStartElement("userid")// for tagging the first parameter
        XMLWrite.WriteString(userid)// setting parameter value
        XMLWrite.WriteEndElement()
            ...
            ...    // code to be written for other parameters
            ...
        XMLWriter.WriteStartElement("fontsize")
        XMLWrite.WriteString(fontsize)
        XMLWrite.WriteEndElement()
    XMLWriter.WriteEndElement()
]
```

Fig 5. UI Specification Generation Algorithm

```
<html>
  <script type="text/javascript">
  function run() {

    xmlhttp=new XMLHttpRequest();
   xmlhttp.open("GET",XML file name,false);
   xmlhttp.send();
    xmlDoc=xmlhttp.responseXML;

    // work out tagNames and grab data from inside each tag
    var x=xmlDoc.documentElement;
    var realChilds=0;
    var parent=x;
    var child=parent.childNodes.length;
    var i = 0;
    var nameofNode="";
    var itemsinNode;
    while(i < child){
        if(parent.childNodes[i].nodeType != 3){
            realChilds++;
            nameofNode=parent.childNodes[i].nodeName;
            itemsinNode=itemsinNode+"|"+
            xmlDoc.getElementsByTagName(nameofNode)[0].childNodes[0].nodeValue;
            }
        i++;
    }
        // print all variables to the document title
     window.document.title = itemsinNode;
    }
  </script>
 </head>
<body onLoad="run();">
</body>
</html>
```

Fig 6. XML Parsing Algorithm

VIII. PERFORMANCE ANALYSIS

We define the above algorithm as the two-step process: the first step is generated of UI specification in XML format; and the second step is implementation of XML data in interface design. We mainly analyze the performance of the two-step process from the availability and the scalability. Other performance, such as security and reliability is relative to the application.

First, we discuss the availability. The algorithm mentioned in this paper is implemented in any client application since the final output is XML. It can be implemented using rich clients such as laptops and PC, or thin clients such as mobile phones and so on. The above algorithm is available and adaptive by context aware.

Second, we talk about the scalability. The algorithm mentioned is implemented for single or multi user. The algorithm is scalable if the application which receives the XML file, adopt the mapping service between receiving XML data and their own data. The service result can be cached to improve the performance of the later use.

IX. CONCLUSION

In this paper, we propose a two-step algorithm for implementing accessible services for VI user using E-Cloud service. We generate the XML file using our e-application. We then place the XML file manually in the E-Cloud service. We then develop the touch screen application to implement the user interface design. And also, evaluate the whole process with different case studies. Since pages are limited to this paper, we didn't elaborate our case studies.

Our work takes a step in this direction, suggesting the accessibility service for VI user using cloud computing. This technique will reduce the burden of novice developers to implement accessibility features for VI users. If these techniques are suitably included, the system usability will be improved. These will advance VI users to utilize the e-learning resources kept in the E - cloud.

REFERENCES

[1] M. M. Rana, U. Rana and M. Fakrudeen, "Cloud Computing Framework and Services Model for the Visually Impaired," International Journal of Technology, Knowledge and Society, vol. 7, no. 2, pp. 1-10, 2011.

[2] M. A. H. Masud and X. Huang, "ESaaS: A New Education Software Model in E-learning Systems," in International Conference, ICCIC 2011 Proceedings, Part V, Wuhan, China, 2011.

[3] S. Ouf, M. Nassar, M. Amr and M. Mossad, "Business Intelligence in the Cloud," in IEEE 3rd International Conference onCommunication Software and Networks (ICCSN), Xi'an, 2011.

[4] P. Pocatilu, F. Alecu and M. Vetrici, "Measuring the efficiency of cloud computing for e-learning systems," WSEAS Transactions on Computers, vol. 9, no. 1, pp. 42-51, 2010.

[5] M. Al-Zoube, "E-Learning on the Cloud," International Arab Journal of e-Technology, vol. 1, no. 2, pp. 58-64, 2009.

[6] B. Wheeler and S. Waggener, "Above-capus services: Shaping the promise of cloud computing for higher education," EDUCAUSE Review, pp. 52-67, 2009.

[7] J. Deyenberg, "Cloud Based Universal Design for Learning Tools," 2013. [Online]. Available: http://www.trailsoptional.com/2013/12/cloud-based-universal-design-for-learning-tools/. [Accessed 05 03 2014].

[8] R. Kwan, R. Fox, F. T. Chan and P. Tsang, Enhancing Learning Through Technology: Research on Emerging Technologies and Pedagogies, World Scientific, 2008.

[9] Y. S. Lee, S. W. Hong, T. L.Smith-Jackson, M. A. Nussbaum and K. Tomioka, "Systematic evaluation methodology for cell phone user interfaces," Interacting with Computers, Volume 18(2), pp. 304-325, 2006.

[10] S. Scutter, "Is Podcasting an Effective Resource for Enhancing Student Learning?," in Teaching Arts and Science with the New

Social Media, Cutting-edge Technologies in Higher Education, Emerald Group Publishing Limited, 2011, pp. 283-295.

[11] S. Marstona, Z. Lia, S. Bandyopadhyaya, J. Zhanga and A. Ghalsasib, “Cloud Computing - The Business Perspective,” Decision Support Systems, vol. 51, no. 1, p. 176–189, 2011.

ISBN: 978-988-19252-7-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)