Accommodating Paper in Document Databases

Majed AbuSafiya

Computer Science Dept
New Mexico Tech
Socorro, NM 87801 USA
majed@nmt.edu

Subhasish Mazumdar

Computer Science Dept
New Mexico Tech
Socorro, NM 87801 USA
mazumdar@nmt.edu

ABSTRACT

Although the paperless office has been imminent for decades, documents in paper form continue to be used extensively in almost all organizations. Present-day information systems are designed on the premise that any paper document in use will be either converted into electronic form or merely printed from electronic file(s) accessible to the system. Yet, paper is the medium of choice in many situations, mainly owing to its portability and usability, and the medium of necessity in others, especially where external communication or the traditional notion of authenticity are involved. Humans who find unique attractive features in both paper and electronic forms of documents, must survive this tension between the de-jure banishment of paper and its de-facto prevalence. In this paper, we propose to make paper documents first-class citizens by including them in the model underlying the information system. Specifically, we extend the schema of a document database with the notion of paper documents, physical locations, and the organizational hierarchy. This leads to an overall enhancement of document integrity and the ability to answer queries such as "where are the customer complaint letters we have received today?" and "which documents are in this filing cabinet?". Recent technological advances such as sensors have made the implementation of such a model very realistic.

Categories and Subject Descriptors

H.4.1 [**Information Systems Applications**]: Office Automation; H.1.m [**Models and Principles**]: Miscellaneous; I.7.1 [**Document and Text Processing**]: Document and Text Editing—*Document Management*; I.7.5 [**Document and Text Processing**]: Document Capture

General Terms

Design, Management

DocEng'04, October 28–30, 2004, Milwaukee, Wisconsin, USA.

Keywords

Document Databases, Document Management, Paper Documents, Paper Manifestation, RFID, Enterprise Document Model

1. INTRODUCTION

Great advances in electronic information technology have made the creation, storage and flow of electronic documents not only feasible but economical, and consequently have led to great increases in productivity. Yet, paper documents exist in virtually every office and are involved in most business processes. There are some intrinsic advantages of paper documents over their digital counterparts [4]: they are easier to work with especially when large, they require little technological infrastructure for reading and writing, they are portable, and easier to annotate. Electronic versions of documents, on the other hand, offer superior search, storage, and transfer capabilities.

An important source of paper in an enterprise is the outside world. Customers, suppliers, and stakeholders send information on paper which are examined, filtered (i.e., some are discarded), routed, and acted upon in paper form before some subset is converted into digital files. Within the organization, paper documents result from printouts of files created and stored digitally; such printouts are used not only for perusal but also for annotation, routing, and signature. In addition, a substantial amount of information is sent out from the enterprise to the external world in paper form. Some documents are deemed to be authentic only in paper form: they are *certificates* or *proofs* of some event, claim, or promise; hence, they must be archived in paper.

Computerized information systems have typically presumed that all important documents are in electronic form. This has resulted in a second-class citizenship for paper documents in the sense that operations on them are invalid until and unless they are coerced into electronic form by scanning, typing, or keying. The belied premise that paper documents will go away has led to their uncomfortable co-existence with electronic document repositories. For the workers, the consequence is tension and inefficiency as they strive to live a double life both among official electronic documents and an unofficial world of paper documents. Our solution is to make paper documents first-class citizens in the document database of the enterprise.

In order to recognize the de-facto status of paper, a simple solution is to keep track of paper documents on a computer database. But managing and accounting for paper documents is hard. First, they are mobile: they move from one employee to another and from one office to another in the organization. Second, paper documents come in various shapes and sizes and, when accumulated, make searching quite difficult. Traditionally, these problems were dealt with by aggregating papers in folders, indexing, and storing them in file cabinets. But even indexing becomes very hard when we have a very large collection of documents and cabinets along with a heavy flow of new documents. Third, operations on paper documents are error prone. Since they are managed and processed by humans, errors are, as expected, quite common: they are misplaced, sent to the wrong person, operations on them are skipped, and they fall into the wrong hands.

We note with great satisfaction some recent technological developments. RFID components (Radio Frequency Identification system) [2, 3, 4] provide a contactless data link between a reader and a transponder. A transponder can transfer data to a reader once it is present within the field of the reader's antenna. The transponder can be passive, i.e., not require any battery power. Typically, a transponder is attached to physical mobile objects in the form of a small cheap tag comprising an antenna, an RF circuit, and some logic circuitry plus memory that may contain an identification number. Such tags may be embedded in paper. For example, a printer that prints on a tag-embedded paper can send back to the database an association between the identifier of the document printed and the tag identifier of the (tagged) paper it was printed on; alternatively, stickers with such tags may be pasted on ordinary paper. Similarly, employee badges can be equipped with a tag containing the employee id number. More advanced transponders may store data obtained from the reader and may have processing abilities. Readers may be attached to desks, file cabinet drawers, and briefcases. Furthermore, movement estimation is also feasible through *attitude microcentrals* [8, 11] which, if embedded within pens, can digitize the act of writing.

These technologies have been the basis for designs of a future office. In [3], it is shown how an office environment can be enhanced with various services including unplanned interactions among employees, and innovative printing facilities. In [4], a computing system is described that allows search of a document based on its tag identifier and provides alerts on delays in document movement. Sensors generate software events that are picked up and processed by an underlying middleware; middleware also stores and handles condition-action rules that are set to specify how the application should react to the various events generated while working with documents.

In this paper, we will exploit these recent sensing technologies in order to propose a model that provides the basis for a solution to the problem of managing paper-based documents. We wish to support queries about the current or last known location of paper documents based on its logical properties (not only its tag id), as well as about a document's access information, and who manages a document. In addition, we would like the ability to answer queries like which paper documents are residing in a location such as a file cabinet, and also ask which documents are missing there. By adopting an enterprise-aware comprehensive model, we can do more than simply search for a document or alert when a particular document arrives. Our model supports integrity constraints that assert that paper documents should be in the right place at the right time as also that they are not under the control of the wrong persons. In addition, it makes it feasible to trigger workflows based on paper documents. In addition to monitoring the check-in and check-out of the paper documents from physical locations, the model also supports the logging of incremental changes applied to those documents; this primarily includes signatures and annotations.

This paper is structured as follows. The next section explains our approach by explaining our proposed model and basic assumptions. The following section outlines how the model is used to attain some of the goals we have listed above. We end with concluding remarks.

2. OUR APPROACH

In this section, we will first enumerate some assumptions and next present our proposed model.

2.1 Assumptions

The feasibility of our approach is based on a number of assumptions. We list the six most important ones here. First, we assume that the technological developments that have already taken place will not be reversed. For example, RFID tags are already extremely cheap to manufacture and install. The motion sensing devices are still expensive but they too should become cheap using economies of scale. Second, we do not assume that every piece of paper need to be identified: only important paper documents, i.e., those that are expected to benefit from tracking, need to be tagged with RFID tags. Third, sensors (readers) need to be placed in file cabinets, desks, and important locations where paper documents are expected to accumulate. Sensors can also be attached to mobile brief cases. Users are to be equipped with RFID badges that can be presented when accessing file cabinets[1]. If it is not important for the enterprise to control access so tightly (as in a university environment), the badge requirement can be ignored. Fourth, the output of the sensors are to be connected to a computer which can process the data stream and update a document database based on our model so that it can answer queries on the documents sensed. A great deal of research is being currently conducted on extending database functionality to process sensor data and manage data streams [6, 7]. Fifth, as a document is sent or routed through different organizational units in the enterprise, sensors can at least pick up when it leaves or enters a department as well as the organization as a whole. Sixth, every paper document has a *manager* who is a person known to the enterprise.

2.2 Proposed Document Model

Our model expands its focus beyond the needs of a standard document database and includes three other related

[1]Badge systems do not necessarily violate privacy requirements[5].

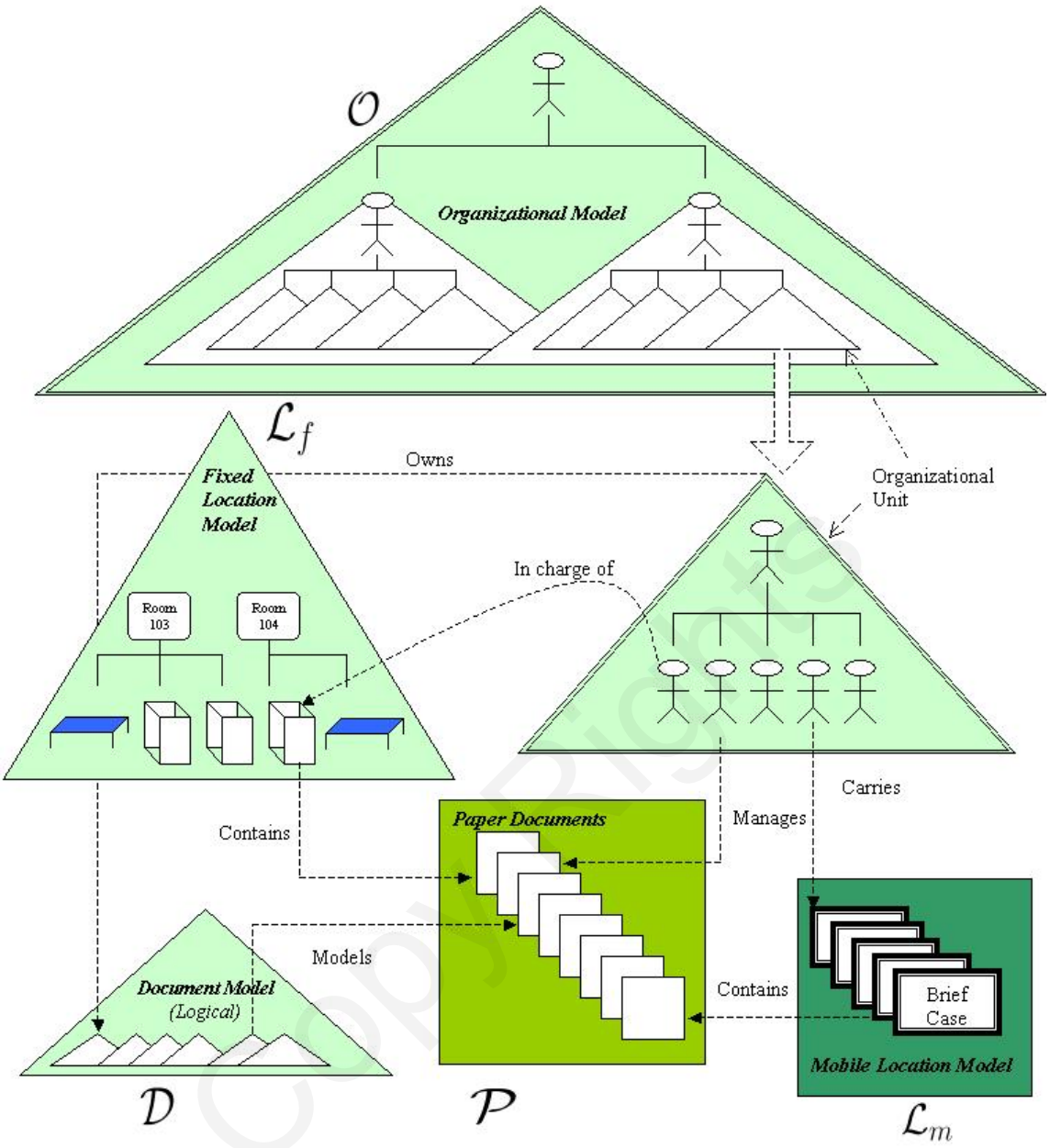

Figure 1: Universal Enterprise Document Model

ingredients of an enterprise: the paper documents themselves, the organizational hierarchy, and the physical locations where documents are stored. While these three ingredients are modeled separately, we view their inter-relationship as crucial for the task at hand. Specifically, we have five such submodels since we need to split the locations into fixed and mobile; we denote them by $\mathcal{O}$, $\mathcal{L}_f$, $\mathcal{L}_m$, $\mathcal{D}$, and $\mathcal{P}$. In the UML (Unified Modeling Language), we would express this in a conceptual level class diagram by representing them via five classes related through associations.

Since the relationships are all one-to-many, we explicitly denote the *many* associations as children. Thus, we obtain a graph model as illustrated in Figure 1 (each arrow from a source node to a destination node implies that the latter is a child potentially with n siblings). Each of the five submodels is modeled as a graph interpreted as semi-structured data [1]. By following these arrows between these submodels, the entire graph can be seen to represent one semi-structured model of the entire document support infrastructure for the enterprise. Moreover, since such semi-structured documents can be implemented using XML [13], the entire model can be expressed as one XML document. We refer to this model as the *Universal Enterprise Document Model*. The word *universal* is used to stress that the model provides an umbrella for every kind of document in an enterprise. The resultant

schema of a document database includes the notions of paper documents, fixed and mobile physical locations, and the organizational hierarchy.

We refer to the model for the standard document database of the enterprise by the *Document Model* ($\mathcal{D}$). We characterize this as a *logical* model since it should typically support the identification of common document components so as to enable the creation of new documents through their reuse via composition and collection. A new document whose parts are old ones is said to be created through *composition*. A folder created by collecting a set of logically related documents is a new document created through *collection*. The folder may not be physical[2]: it may well mean a category of documents. The Document Model $\mathcal{D}$ can be represented by a graph where each node represents a document; if it is created by composition or collection, those constituent documents are connected through child pointers. The leaf nodes become the atomic documents. $\mathcal{D}$ represents the electronically created documents known to the standard information system of the enterprise; as such, we do not presume that our graph-like model be supported directly. In implementing our model it is enough if we can map the nodes of $\mathcal{D}$ to elements of the standard document database of the enterprise. If the latter is as primitive as a filesystem on a PC, the implementation of our model need only remember the filenames and directory structure and have the ability to monitor changes on the files.

There are two models of physical locations: the *Fixed Location Model* ($\mathcal{L}_f$) and *Mobile Location Model* ($\mathcal{L}_m$); they describe the physical locations of the organization that are fixed in space and subject to movement respectively. $\mathcal{L}_f$ is a hierarchical structure covering building, floor, room, etc., down to a desk or file cabinet; the level of detail depends on the granularity of physical locations supported by the enterprise and sensors installed. Consequently, the exact physical location of a paper document within the enterprise can be represented by a directory-path-like address. For example, the physical location of a student application in the university may be
/Univ/AdminBldg/Admission/FileCabinet3/Drawer4/@0023.
$\mathcal{L}_f$ must cover all possible physical locations of documents in the organization that can be sensed. Mobile locations are typically briefcases as they are commonly used to transport paper documents during perusal or review cycles. An interesting mobile location is a ring-binder (a ring-bound folder) containing sensors; a ring-binder typically contains a *coherent* collection of paper documents.

The organizational structure in the enterprise can be described by a model that describes different organizational units and as well as the employees that work for them. We follow the *Organizational Model* proposed by the Workflow Management Coalition [14] in which the enterprise is composed of organizational units that *supervise* other organizational units. If we view the organizational structure as a tree of nodes, we find that the nodes are organizational units, a higher node (or organizational unit) in this hierarchy is a supervisor of a lower node (or organizational unit). Our *Organizational Model* ($\mathcal{O}$) consists of both organizational units and humans who play certain roles.

The Paper Model ($\mathcal{P}$) is intended to capture all paper documents (that are important enough to be tagged and sensed). Treating a paper document as semi-structured data has the advantage that it captures what is currently known about it allowing other attributes to be added as they become known[3]. This is very important for documents that come from external sources as it allows them to be routed and processed incrementally. In fact Query 6 above assumes that a paper student application has been scanned by an admission clerk who has keyed in on her PC an attribute called GPA; this attribute has been added to the document node in $\mathcal{P}$. A paper document is a leaf node in $\mathcal{P}$. $\mathcal{P}$ also contains collection nodes that represent a collection of leaf nodes, i.e., a folder of actual paper documents. Such a collection node however can only contain leaf nodes, i.e., not collection nodes (a folder cannot contain another folder). In addition, there can be composition nodes in $\mathcal{P}$; they represent a document created by piecing together component documents (e.g., a cover sheet stapled to a research proposal).

Figure 1 also shows relationships between $\mathcal{O}$, $\mathcal{L}_f$, $\mathcal{L}_m$, $\mathcal{P}$, and $\mathcal{D}$. First, consider the relationship *owns*. Each organizational unit has a number of responsibilities that distinguish it from other organizational units; this means that each organizational unit deals with documents that are under its responsibility and management. For example, the Registrar's office takes care of students' registration documents while the Admission office takes care of student admission applications. We view the organizational units as departments that deal with and own a certain set of documents. This is captured by the relationship *owns* between $\mathcal{O}$ and $\mathcal{D}$; in a sense, it reflects a document-oriented functionality of the organizational structure. It indicates for every folder and document in $\mathcal{D}$, which organizational node owns it. Each such organizational unit employs humans and gives them roles that define which documents they are authorized to access and which operations they are allowed to perform on the documents managed by the department. Our model's support of paper documents implies a more robust access control as it becomes possible to specify restrictions on physical locations of a paper document and also check that they are consistent with those on the digital version. We are omitting discussion on details of access control.

Both organizational units and humans working for them are placed in charge of fixed physical locations like buildings, rooms, file cabinets, and desks. This is the *in charge of* relationship between $\mathcal{O}$ and $\mathcal{L}_f$.

The *contains* relationship between $\mathcal{L}_f$ and $\mathcal{P}$ indicates that a fixed physical location has a certain physical doc-

[2]We use the term *physical* to distinguish from *virtual*; the access modalities of a paper document or binder differs significantly from that of a digital file or directory: the former is limited to a handful of people in the vicinity whereas the latter can potentially be accessed concurrently by countless agents across the globe. By contrast, in the FRBR[10], the term is associated with the physical embodiment of an *expression*: a process of translation from the intellectual realm into a physical *manifestation*; in that sense, a typeset printed document is as physical as its corresponding digital file on a hard disk.

[3]Of course, it also helps in capturing decomposition into text information objects and graphic information objects.

ument in/on it. Just like the fixed locations, there is a relationship *contains* between mobile locations $\mathcal{L}_m$ and the paper documents $\mathcal{P}$. For example, a folder containing a collection of paper documents is associated with a ring-binder which is its physical location (a ring-binder may contain more than one folder). Mobile locations like briefcases are presumed to be always carried by a person, hence the *carries* relationship.

Every paper document is assigned to an individual (internal or external) who manages it; this is captured by the *manages* relationship. The person who checks out a document from its cabinet becomes its manager. The employee that receives an external document manages it.

3. USING THE MODEL

In this section, we will outline how our proposed model can support useful queries, exploit constraints, and help with operations on paper documents.

3.1 Queries

We will list some queries and show that they can be expressed in XQuery [15].

1. *Which documents are in File Cabinet3 of Room 201 in Brown Hall?*

```
for $t IN doc(L_f.xml)//BrownHall/Room201/
                    Cabinet3/contains/document
 for $d IN doc(P.xml)//document
  where $t.id=$d.id
  return
   <Result>
    $d
   </Result>
```

The result is of the form:

```
<Result>
 <document> .... </document>
</Result>
<Result>
 <document> .... </document>
</Result>
<Result>
 <document> .... </document>
</Result>
```

The // means that the element that follows it may be nested anywhere within the preceding element.

2. *Where is the budget proposal?*

Let us suppose that we keep the physical location of paper documents as an element which captures the inverse of the *contains* relationship.

```
for $t IN
 doc(P.xml)//document
 where $t/title='BudgetReport'
 return
  <Result>
   $t/physicalLocation
  </Result>
```

The result is of the form:

```
<Result>
 <physicalLocation>
   ManagementBuilding/PresidentOffice/desk
 </physicalLocation>
</Result>
```

3. *James Hunter's application was not found in file cabinets or desks. Who could be carrying it?*

```
for $t IN doc(O.xml)//employees/employee
 for $d IN doc(P.xml)//document
  where $t/carries/*/contains/document.id = $d.id
    and $d/title='application'
    and $d/name='James Hunter'
  <Result>
   $t/name
  </Result>
```

In the above, the wildcard $\star$ will range over all mobile locations. (It is possible to refine it further so as to restrict it to only mobile briefcases.)

The result is:

```
<Result>
 Tom Will
<Result>
```

4. *Give a list of all applications in the Admission Office from students who have applied to the Computer Science department using a paper form.*

```
for $t IN doc(D.xml)//Admission/Applications/
                                      application
 for $d IN doc(P.xml)//document
  where $t.id=$d.id and
        $d/department = 'Computer Science'
  <Result>
   $d
  </Result>
```

The result will be a list of application documents of the form

```
<Result>
 <document ... >
 :
 </document>
 <document ...>
 :
 </document>
 :
</Result>
```

5. *Give a list of the paper documents that Tom manages.*

Suppose that an *employee* node has a nested *manages* node.

```
for $t in doc(O.xml)//employee[name='Tom']/
                              manages/document
 for $d in doc(P.xml)//document
  where $d.id=$t.id
   return
    <Result>
     $d
    </Result>
```

The result is

```
<Result>
 <document>  .... </document>
 <document>  .... </document>
 <document>  .... </document>
 :
 :
<Result>
```

6. *Give the physical locations of all applications of students with GPA > 3.0.*

```
<Result>
for $b in doc(D.xml)//Admission/Applications/
                                      application
 for $t in doc(P.xml)//document
  where $b.id=$t.id and $t[GPA > 3.0]
   return
    <physicalLocation>
     $t/physicalLocation
    </physicalLocation>
</Result>
```

The result is

```
<Result>
 <physicalLocation>
  AdmBld/AdmissionOffice/Cabinet4/@342
 </physicalLocation>
 <physicalLocation>
  AdmBld/AdmissionOffice/Cabinet2/@512
 </physicalLocation>
 :
 :
 :
</Result>
```

7. *Give a list of names of employees that are currently carrying briefcases along with the id's of those briefcase.*

```
for $b in
 doc(O.xml)//employee
 return
  <employeeBriefCase> {
   for $a in
    $b/carries/briefCase
    return
     <Result>
      $b/name,
      $a/briefcaseID
     </Result>
   }
  </employeeBriefCase>
```

The result is:

```
<employeeBriefCase>
 <Result>
  <name>... </name>
  <BriefCaseID> ... </BriefCaseID>
 </Result>
 :
 <Result>
  <name>... </name>
  <BriefCaseID> ... </BriefCaseID>
 </Result>
 :
<employeeBriefCase>
```

8. *List the titles of documents that are now mobile.*

```
<Result>{
for $b1 in
 doc(L_m.xml)//briefCase/contains/document
 for $b2 in doc(P.xml)//document
  where $b1.id=$b2.id
  return
   <title>
    $b2/title
   </title>
}
</Result>
```

The result is

```
<Result>
  <title>  Budget Report </title>
  <title>  Application Form </title>
</Result>
```

3.1.1 Location Views

Views on a database restrict access to certain data for, among other reasons, privacy and confidentiality. Location views can similarly restrict access to a certain granularity in location detail for a class of users. For example, the R&D department may need to know that their proposal has reached the Finance department but should not be aware of exactly which desk it is sitting on; that detail should be available only to personnel in the Finance Department. Such location views can be easily implemented by associating a cutoff node N in $\mathcal{L}_f$ for each user; if a query returns a result node below N, it is replaced by N.

3.2 Constraints

Constraints on the Universal Enterprise Document Model can be used to enhance the integrity of paper documents. While it is not feasible, at this stage of technological development, to prevent the violation of integrity, even the detection of its violation is useful as we can thereby alert appropriate personnel and often take corrective measures.

One class of constraints state that every element of $\mathcal{P}$ *exists*, i.e., can be located by some sensor until such time when it has been explicitly destroyed by its owner. In practice, it is enough to verify the existence periodically and allow for time gaps when a document is routed from one location

to another. Another class of constraints deals with the integrity of a collection of paper documents, i.e., whether or not a folder stored in a ring-binder is missing a constituent document.

Constraints are usually associated with the model when it exhibits multiple paths from one node to another (not necessarily cyclic). For example, consider $\mathcal{O}$ and $\mathcal{P}$: there is one path via $\mathcal{L}_f$ and and another directly; sure enough, there is an associated constraint: we should alert a person in $\mathcal{O}$ *in charge of* a desk in $\mathcal{L}_f$ that *contains* a paper document in $\mathcal{P}$ but has not been assigned to *manage* it.

Typically, there are temporal constraints between $\mathcal{D}$ and $\mathcal{P}$. For example, a document in $\mathcal{D}$ is printed as a paper document in $\mathcal{P}$, with the promise of checking back the changes on paper into $\mathcal{D}$; it may be important to know if either the original document in $\mathcal{D}$ or the paper version in $\mathcal{P}$ has been changed so that an update does not create a stale copy. This is the well-known data replication problem and various strategies have been offered, for example leases [9]. Straightforward checkin-checkout schemes for version control are supported by current commercial systems like the Lotus Domino Document Manager (IBM) and DocuShare (Xerox). However, there are some subtleties with version control of paper documents. Successive annotations on a paper document preserves the version whereas updates on a digital file typically results in different versions. Thus at the very least, we need a map from $\mathcal{D}$ to $\mathcal{P}$ representing versions. We will report elsewhere on the ramifications of version control.

3.3 Operations on Paper

A very important operation on paper documents is annotation. When notes are written in the margin of a paper document that was obtained from a digital file in $\mathcal{D}$, it diverges from that electronic version. Usually, at some point, it is desirable that the annotations be available in digital form. First, as we have mentioned earlier, technological advances in motion estimation have resulted in pens that can transmit the atomic movements to a PC from which a graphic file can be created storing those annotations. Second, it is possible to eventually scan the modified paper document. Third, though the least desirable, it is possible to key in the annotated text. In the context of our model, we note that even if none of these are feasible, it is extremely beneficial and often adequate for a sensor to learn that a paper document has been modified. This would require a simple action such as a hole to be punched in a corner that would change the bit pattern transmitted by the transponder. This is useful because the fact that the paper document has been changed can now be communicated by the system to any other personnel requesting to read the original document in $\mathcal{D}$. If the annotation needs to be shared, the system will generate a routing procedure for the annotated paper document or a copy after verifying the access rights. In general, the constraints between the element in $\mathcal{D}$ and the one in $\mathcal{P}$ will guide future requests for access to the original in $\mathcal{D}$ and further copies of the paper document.

Certain operations on paper documents can take advantage of some assistance from a system based on our model. For example, every manager knows of a situation where a folder containing some important documents had to be quickly assembled by cannibalizing from existing folders. In these situations, assembly time is the key and digital files are not an option either because a document was never available in digital form or because the document will take too long to print, or that the scribbled annotations on paper are crucial. By monitoring and storing the contents of ring-binders, our system can guide this process of cannibalization where documents are removed from different binders and a new binder assembled hastily and yet accurately. The even better news is that the system will never forget the constraints on folders and consequently will send unceasing reminders that the original folders need to be re-assembled.

Paper documents move from office to office, from one employee to another. The movement of tagged documents can be monitored and (as we see in the next subsection) initiated by the system. Once the destination is known, the system can simply alert a secretary that a certain document on a desk needs to be sent to a certain department. The same applies to the documents leaving the organization for the outside world. In fact, the inter-office mail system can be highly automated by eliminating writing addresses on envelops and allowing the system to determine the destination of tagged documents.

3.4 Augmenting Workflows

Workflow systems can now be made aware of paper documents. Paper-based workflows, which exist unofficially today, can take advantage of automation and be more reliable in the system we envision.

Consider the situation when a paper document arrives from a customer, an external source, in the mail. A secretary opens the mail, puts a tagged sticker on it,[4] and enters on the computer the category of the mail it is: a customer complaint letter. At this point, the paper document has an entry in $\mathcal{P}$ with only an identification number and a category. The system has a pre-defined workflow system dealing with customer complaints: it needs to be routed to the Customer Service clerk. The system alerts secretaries, mail persons about this document that needs to be sent off to the destination desk. Of course, the system can also decide and inform the workflow participant where the document should go next. As is standard, the workflow can be designed in various ways including Event-Condition-Action (ECA) rules.

4. CONCLUSION

It is important to stress what we are *not* arguing. We are not resisting the ideal of a paperless office, and we are definitely not claiming that paper is superior to the digital form. We are only accepting the reality that paper continues to co-exist with digital documents in spite of decades of technological developments; this work is aimed at mitigating the consequent complications for the workers and inefficiency for the enterprise. The reason for this co-existence is that each form has exclusive advantages and limitations, some of which we have enumerated in the introduction. We are also not predicting that this state of affairs will last for-

[4]There is also the chance that the document that has just arrived is already tagged. We assume for simplicity that there are no conflicts between external and internal tags.

ever: some future technological development may really and truly eliminate our dependence on paper. But until then, we want to ensure a harmonious co-existence of paper and digital files by giving workers who deal with paper the benefit of database technology.

The current approach toward paper document is to scan them into digital form, and print them out when necessary. This works well for checks in a bank (customers still prefer to write paper checks). But it does not work well with non-standard input, e.g., letters from customers with odd-shaped attachments, because of the labor involved. Also, it is labor-intensive to carry out the scan-print cycle after every annotation on a document being routed [5]; and it is wasteful when the document is multi-page (most of the document may have to be reprinted). The use of Optical Character Recognition is seriously error-prone when handwriting is involved; as a result, scanning creates an image but this undermines the efficacy of search, the hallmark of digital documents. At a high level of abstraction, scanning and using tags may both be viewed as the application of technology to paper. But there are major practical differences: sticking an RFID tag is much easier and needs to be done once; it is not comparable with repeated scanning.

To summarize, in this paper we have argued that paper documents should be first-class citizens in a document model for an enterprise reflecting its de-facto status. Toward that end, we have proposed a Universal Enterprise Document Model. Our model is feasible to implement given recent technological developments in sensor technology and sensor data management. We have outlined how this model can be used to answer useful queries about the location of particular paper documents (the queries are based on their logical attributes, not merely on their identification numbers) as well as about the contents of paper repositories on desks or file cabinets or briefcases. The model helps the designer articulate various constraints that promote the integrity of the paper documents whose improved monitoring facilitate various practical operations including workflows.

For future work, we propose to add details to the model to support copies, versions, and access control. Further, we are working on the integration of the model with the business process management of the enterprise [12].

5. ACKNOWLEDGMENTS

The authors are grateful to anonymous referees whose critical comments have greatly improved this paper.

6. REFERENCES

[1] S. Abiteboul, P. Buneman, and D. Suciu. *Data on the Web: From Relations to Semistructured Data and XML*. Morgan Kaufmann, 2000.

[2] AIM Global Network. Radio Frequency Identification. http://www.rdif.org.

[3] J.-M. Andreoli, S. Castellani, A. Grasso, J.-L. Meunier, M. Muehlenbrock, J. O'Neill, F. Ragnet, F. Roulland, and D. Snowdon. Augmenting Offices with Ubiquitous Sensing. In *Proc. Smart Object Conference, Grenoble, France*, 2003.

[4] D. Arregui, C. Fernstrom, F. Pacull, G. Rondeau, J. Willamowski, E. Crochon, and F. Favre-Reguillon. Paper-based Communicating Objects in the Future Office. In *Proc. Smart Object Conference, Grenoble, France*, 2003.

[5] A. Beresford and F. Stajano. Location Privacy in Pervasive Computing. *IEEE Pervasive Computing*, pages 46–55, Jan–Mar 2003.

[6] P. Bonnet, J. Gehrke, and P. Seshadri. Towards Sensor Database Systems. In *Proc. 2nd International Conference on Mobile Data Management*, pages 3–14, 2001.

[7] D. Carney, U. Getintemel, M. Cherniack, C. Convey, S. Lee, G. Seidman, M. Stonebraker, N. Tatbul, and S. Zdonik. Monitoring Streams — A New Class of Data Management Applications. In *Proc. Very Large Data Bases*, pages 215–116, 2002.

[8] D. Fontaine, D. David, and Y. Caritu. Sourceless Human Body Motion Capture. In *Proc. Smart Object Conference, Grenoble, France*, 2003.

[9] C. G. Gray and D. Cheriton. Leases: An Efficient Fault–Tolerant Mechanism for Distributed File Cache Consistency. In *Proc. 12th ACM Symp. on Operating Systems Principles*, pages 202–210, 1989.

[10] IFLA Study Group on the FRBR. Functional Requirements for Bibliographic Records: Final Report, 1998. Deutsche Bibliothek, UBCIM Publications N.S., Vol. 19.

[11] C. Joguet, Y. Caritu, and D. David. Pen-like, Natural Graphic Gesture Capture Disposal, Based on a Microsystem. In *Proc. Smart Object Conference, Grenoble, France*, 2003.

[12] S. Mazumdar and M. AbuSafiya. A Document-Centric Approach to Business Process Management. In *Proc. Intl. Conf. on Information and Knowledge Engineering*, pages 461–466, 2004.

[13] J. McHugh, S. Abiteboul, R. Goldman, D. Quass, and J. Widom. Lore: A Database Management System for Semistructured Data. *SIGMOD Record*, 26:54–66, Septemper 1997.

[14] Workflow Management Coalition. Workflow Management Coalition Terminology & Glossary, 1999. http://www.wfmc.org/standards/docs/TC-1011_term_glossary_v3.pdf.

[15] World Wide Web Constortium. XML Query (XQuery). http://www.w3.org/XML/Query.

[5] Of course, labor is cheap in large parts of the world, but scanning and eliminating paper may be counter-productive there since a PC on every desk is hardly realistic.