Design of a Modular Architecture for Integrating Paper and Digital Document Management Systems

Matthew Jervis
Department of Computer Science
The University of Waikato
Hamilton, New Zealand
mjervis@cs.waikato.ac.nz

Masood Masoodian
Department of Computer Science
The University of Waikato
Hamilton, New Zealand
masood@cs.waikato.ac.nz

ABSTRACT

Paper documents continue to play an important role in many of today's offices. This leads to the coexistence of both paper and digital documents, with each type typically being managed using completely separate systems. There is therefore a need for systems that bridge the gap between the two realms of paper and digital document management. We have developed a paper document management system, called SOPHYA, which provides mechanisms for a seamless integration with conventional digital document management systems. This paper describes the modular architecture of SOPHYA, and demonstrates its flexibility in supporting development of different types of client applications.

Author Keywords
Document management, paper documents, digital documents.

ACM Classification Keywords
H4.1 Information systems applications: Office Automation;
H3.6 Information storage and retrieval: Library Automation

General Terms
Design, Human Factors, Management.

INTRODUCTION

Paper documents have a number of affordances that are not provided by digital documents; such as fast and flexible navigation, ability to read over multiple documents at once, etc. [9]. Until digital alternatives to paper can provide these affordances, it is likely that paper documents will continue to be used along with digital documents. It is therefore crucial to develop document management systems (DMS) which integrate organisation of digital and physical documents in a seamless manner to support better document workflow.

Various systems have been developed for tracking of physical documents, while providing some degree of connectivity with existing digital DMS. Most of these systems use RFID tags to augment real-world artefacts. One such system by Arregui et al. [2] uses RFID readers at various locations around the office to scan and track tagged documents. This system has been deployed in a patent office by O'Neill et al. [6]. Hark et al. [3] also describe a similar system, in which RFID tags are printed directly onto documents using e-ink technology. Other examples include research by AbuSafiya and Mazumdar [1] who propose a model that incorporates paper documents into a digital DMS using RFID, and Raskar et al. [7] who use active RFID tags, augmented with photosensing capability, in conjunction with a handheld projector that when aimed at the tags is able to determine their relative location and project visualisations onto them.

We have developed several alternative versions of a DMS called SOPHYA, which has been more fully described elsewhere [8, 4, 5]. SOPHYA is a technology for augmenting paper DMS to allow them to be integrated with digital systems. Unlike other systems referred to here, SOPHYA utilises wired communication rather than RFID for tracking physical documents. A wired system allows power to be supplied to electronic components attached to physical artefacts, so that they can provide more advanced functionality. This is clearly not possible with passive RFID systems, and active RFID systems require the use of batteries.

A full discussion of the implementation details of SOPHYA is beyond the scope of this paper. Here we describe the modular architecture of SOPHYA, and demonstrate its effectiveness in providing the necessary mechanisms for development of client applications which can be connected to SOPHYA to allow management of physical documents, while also integrating with conventional digital DMS used for organisation of electronic documents.

ARCHITECTURE OF SOPHYA

Figure 1 shows the architecture of SOPHYA, which is split into five layers: three of them belonging to SOPHYA (hardware, firmware and middleware components), and two that are application specific (digital DMS server, and the clients).

The hardware component manages the *containers*, which hold a collection of documents (e.g. folder, archival box) rather than the individual documents themselves. Containers can be placed in *physical storage locations* (e.g. filing cabinets, shelves, in-trays), which can be either ordered [5], or unordered [4]. Both the container and physical storage lo-

UbiComp '10, Sep 26-Sep 29, 2010, Copenhagen, Denmark.
ACM 978-1-60558-843-8/10/09.

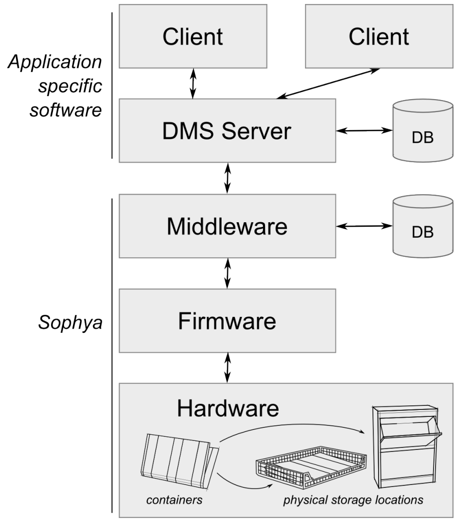

Figure 1. Overview of the architecture of SOPHYA.

cation are augmented with electronic circuitry. These components will be different depending on the type of SOPHYA technology used (e.g. ordered or unordered). However, each container would have a unique ID, and may have optional user interface components such as LEDs.

The firmware is the software embedded into each physical storage location which gives it a unique ID, and allows it to communicate with the containers (e.g. read their IDs and control their user interfaces), as well as communicating this type of information to the middleware.

The middleware, on the other hand, is responsible for dealing with data coming from different physical storage locations attached to SOPHYA, and presents this information to any application specific DMS server using it. The middleware maintains its own database to keep track of information related to individual containers (e.g. their IDs, location).

It is important to note that within the internal layers of SOPHYA there is no concept of digital documents. All that these layers are aware of is that there are a series of containers and locations, each with their own unique ID, and where each container is located. This type of information can, however, be queried by the application specific software which is then responsible for mapping it to digital content it is dealing with. This separation of the physical management of documents from the digital management of content associated with those documents makes it possible for a range of client applications to be developed. The separation also makes it possible for the internal components of SOPHYA to be modified without requiring the modification of the application specific software. We have already developed two versions of SOPHYA, ordered [5] and unordered [4], which can interact with application specific software seamlessly.

SOFTWARE INTEGRATION

As mentioned earlier, SOPHYA has been designed to facilitate integration of application specific DMS and clients with its middleware component. DMS software can connect to the middleware to receive information about the physical storage locations and containers. Software connecting to the middleware can opt to subscribe and receive events (e.g. containers added or removed), as well as being able to query for specific information when required.

The interface between the middleware and the application is, by design, abstracted from the hardware for reason of modularity. The type of SOPHYA hardware used in a particular setting would be dependent on the physical document management requirements of that specific setting. For instance paper documents storage and retrieval needs of a small law firm are radically different from the requirements of a library that needs to handle a large number of books; and as such the type of SOPHYA hardware used in each of these settings would be different. In a law firm it would be sufficient to know where a document is, and so an unordered system [4] might be all that is needed. In a library, on the other hand, it is also important to know the physical order of the books on shelves, what is before and after a book, etc., so a more advanced ordered system [5] would be more suitable.

Therefore, the information that can be queried from the middleware may not be possible with all hardware configurations, and as new hardware platforms are developed more queries may be supported by the appropriate middleware. However, an important requirement is that all different types of middleware support a set of basic functionality (i.e. event notification and queries) and they degrade gracefully when a given query is not supported by a specific hardware. The following sections describe these basic events and query types.

Events

There are a number of cases where the DMS software may want to be notified of events by SOPHYA. For example, when a folder is placed in a user's physical in-tray an alert could be sent to their email if they are out of the office. Currently there are two possible events, addition or removal of containers, to which a DMS can subscribe. The application can receive notification when a container is added to, or removed from, specific locations or across the whole system.

Queries

To allow the application to get information about the locations and containers, the middleware provides a virtual database which the application software is able to query. This is divided into three virtual tables, one for locations, one for containers and one for event history. These are virtual in that they do not exist in an static sense, but rather the information is gathered dynamically when a query is received. Table 1 lists the fields provided by each table.

This virtual database gives access to all of the information SOPHYA is currently able to provide about the physical artefacts it manages. By querying this database it is possible to answer questions such as: *Where is container x located? Which containers are at location y? Which containers are not currently present at a location? Which containers have been moved since time z? How long has container m been at*

Physical Storage Locations	
ID	the unique ID of the location.
Type	the type of location (e.g. filing cabinet, document tray, etc.).
Last Event	the ID of the most recent event at this location.
Last Accessed	the time of the most recent event at this location.
Containers	
ID	the unique ID of the container.
Type	the type of container (e.g. folder).
Location	the ID of the current location of the container (or null if it is not currently present in the system).
Position	the current position of the container in its present location (if available).
Last Event	the ID of the most recent event this container was involved in.
Last Accessed	the time of the most recent event this container was involved in.
Events	
EventID	the unique ID of the event.
Type	the type of event (e.g. added or removed).
Time	the time at which the event occurred.
ContainerID	the container which this event involved.
LocationID	the location which this event involved.

Table 1. Virtual tables provided by the SOPHYA middleware.

its current location? Which containers are in the vicinity of container n? Where has container i been?

EXAMPLE APPLICATIONS

SOPHYA provides low level information about document containers (e.g. folders, books) and their location, which then needs to be processed by application specific software designed to support the requirements of a particular setting. As mentioned earlier, such application software is likely to integrate with conventional digital DMS. To demonstrate SOPHYA's flexibility in this respect, we provide two demonstrative examples of such applications.

Library Application

Libraries are a good example of a scenario where management of physical artefacts (e.g. books, journals) can benefit from integration with digital systems. Most libraries have a combination of digital cataloguing and lending systems, which are separate from the mechanism used for storage and retrieval of items; the only link being the cataloguing label attached to individual items. Because of the lack of systems for digital tracking of artefacts, or the lack of sufficient integration between the digital and physical management systems, most large libraries often have problems with items being misplaced or lost permanently.

In this particular case an integrated digital library system can be developed which would rely on an ordered version of SOPHYA [5] for physical tracking of library items in terms of location, ordering on shelves, etc., while being linked to the cataloguing and lending systems. Figure 2 shows the architecture of such a system. On the hardware layer, items such as books are augmented with container circuitry, and shelves are augmented with physical storage location circuitry. Their firmware then allows them to communicate with the SOPHYA middleware, which maintains its internal database as described previously.

The library management system, on the other hand, is responsible for getting items' location information from SOPHYA, and cataloguing and lending information from the existing library system, and making them available to client software. The *loan database*, keeps tracks of items that are on loan, the *catalogue database* provides catalogue information about the items (e.g. title, author, etc.), and SOPHYA's middleware provides dynamically changing information about the current location of items in the library.

Clients access integrated library collection information through the library management software. There may be different clients for different purposes, allowing different levels of access. For example, the librarian client would allow librarians to access loan information for all library patrons and add entries to the catalogue database, while the patron client would only be able to view information about books and loan information for the logged-in user. Other clients could include on-shelf visualisations, remote browsing, etc.

Design Office Application

Another example of a typical kind of office document workflow that could benefit from the integration of physical and digital document management systems is demonstrated using the following fictitious scenario, which is actually based on our observation of an existing graphic design office:

The workflow begins with sales representatives in the field getting jobs from clients. A "job sheet" is then filled in for each job, and is brought back to the office along with any related hardcopy material. All material relating to a job is placed in a "job-bag". Information from the job sheet is then entered into the job management software and the job is assigned a unique ID. The job-bag goes to whoever is working on the job, and may get passed around if more than one person needs to work on it, though only one person can have it at a time. Job-bags for jobs that are incomplete, but not currently being worked on are stored on shelves at the centre of the office. These shelves provide a quick visual indicator of how much work remains to be done.

This workflow process could be supported using an unordered version of SOPHYA [4], as shown in Figure 3. The physical in-trays and desktops of the designers are augmented with physical storage location circuitry, and the job-bags become containers. Other than the physical artefacts being managed, SOPHYA would function in the same way as in the library example.

The differences appear on the application specific software side. In this case SOPHYA is integrated with existing job management system currently used by the example design

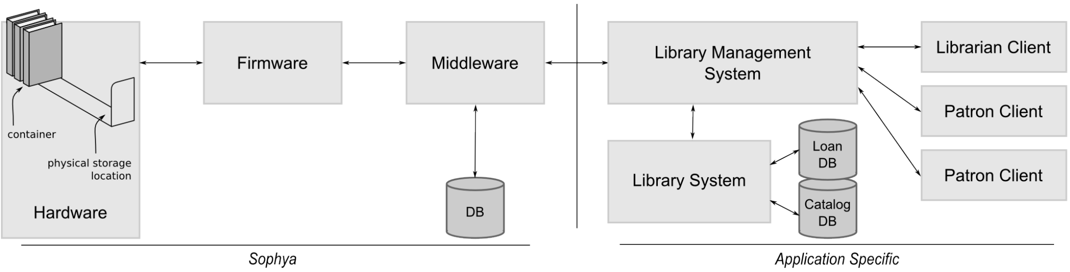

Figure 2. Library application.

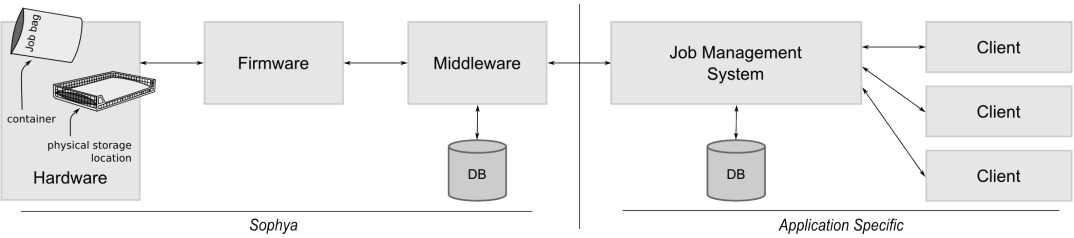

Figure 3. Design office application.

office. As each job (and thus job-bag) is already assigned a unique ID, it is simply a matter of creating a mapping between the job ID and the container ID of the augmented job-bag. When entering the digital information about a job into the job management system, the sales representative would place the job-bag on a container reader (similar to an in-tray) to create a mapping between its job ID and container ID.

A more advanced job management system and client software can then be developed to provide valuable information on tracking job-bags as they are processed and moved between different people, making it possible to dynamically view job-bag location, history, digital content associated or needed, etc.

CONCLUSIONS

This paper described the modular architecture of SOPHYA, which has been designed to facilitate seamless integration with conventional digital DMS. The demonstrative application software discussed in this paper illustrated how this integration can be achieved in two radically different case scenarios. We are currently in the process of developing a prototype library management and visualisation client software.

REFERENCES

1. M. AbuSafiya and S. Mazumdar. Accommodating paper in document databases. In *DocEng '04: Proceedings of the 2004 ACM symposium on Document engineering*, pages 155–162, New York, NY, USA, 2004. ACM.

2. 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 conf., Grenoble, France*, 2003.

3. B. S. Hark, J. H. Choi, and C. S. Leem. A database design of rfid document management system with e-ink technology. In *NCM '08: Proceedings of the 2008 Fourth International Conference on Networked Computing and Advanced Information Management*, pages 14–17, Washington, DC, USA, 2008. IEEE Computer Society.

4. M. Jervis and M. Masoodian. Digital management and retrieval of physical documents. In *TEI '09: Proceedings of the 3rd International Conference on Tangible and Embedded Interaction*, pages 47–54, New York, NY, USA, 2009. ACM.

5. M. G. Jervis and M. Masoodian. Sophya: a system for digital management of ordered physical document collections. In *TEI '10: Proceedings of the fourth international conference on Tangible, embedded, and embodied interaction*, pages 33–40, New York, NY, USA, 2010. ACM.

6. J. O'Neill, A. Grasso, and J. Willamowski. Rfid: enhancing paper documents with electronic properties. In *CSCW, Computer Supported Cooperative Work, Workshop Collaborating Over Paper and Digital Documents*, 2006.

7. R. Raskar, P. Beardsley, P. Dietz, and J. van Baar. Photosensing wireless tags for geometric procedures. *Commun. ACM*, 48(9):46–51, 2005.

8. T. Seifried, M. Jervis, M. Haller, M. Masoodian, and N. Villar. Integration of virtual and real document organization. In *TEI '08: Proceedings of the 2nd international conference on Tangible and embedded interaction*, pages 81–88, New York, NY, USA, 2008. ACM.

9. A. J. Sellen and R. H. Harper. *The Myth of the Paperless Office*. MIT Press, Cambridge, MA, USA, 2003.