Web-Based Integration of Printed and Digital Information

Moira C. Norrie and Beat Signer
{norrie, signer}@inf.ethz.ch

Institute for Information Systems
ETH Zurich, CH-8092 Zurich, Switzerland

Abstract

The affordances of paper have ensured its retention as a key information medium, in spite of dramatic increases in the use of digital technologies for information storage, processing and delivery. Recent developments in paper, printing and wand technologies may lead to the widespread use of digitally augmented paper in the near future, thereby enabling the paper and digital worlds to be linked together. We are interested in using these technologies to achieve a true integration of printed and digital information sources such that users may browse freely back and forth between paper and digital resources. We present web-based server technologies that support this integration by allowing users to dynamically link areas of printed documents to objects of an application database. The server component is implemented using the eXtensible Information Management Architecture (XIMA) and is independent of the particular paper, printing and reader technologies used to realise the digitally augmented paper. The framework presented manages semantic information about application objects, documents, users, links and client devices and it supports universal client access.

1 Introduction

The issue of architectures and technologies for the integration of existing data sources has been a topic of interest for many years within the database community. A number of solutions have been proposed with the approach taken dependent on the degree of heterogeneity, the level of local autonomy and, of course, the required functionality of the target system. In terms of heterogeneity, many forms of data sources have been covered including different categories of database management systems, knowledge base systems, file systems and web documents. But one major form of data source has so far been neglected, namely paper documents.

Predictions of the paperless office no longer seem realistic as indicated by the continued increase in office paper consumption over the last two decades [22]. Digital technologies have advantages in terms of storing and accessing large amounts of information, displaying multimedia and fast full-text searching. Further, it is a dynamic medium that enables both contents and links to be updated with ease. But paper technology has its own advantages over digital media in terms of navigation within and across documents. It is much simpler to scan a book by rapidly flicking through pages than to browse through a digital document. Paper also supports forms of collaboration and interaction that are difficult to mimic in current digital environments. Ideally, we would like to achieve an integration of paper and digital technologies, thereby gaining the best of both worlds.

While a number of projects have proposed different variants of digitally augmented paper as a means of linking printed and digital media, the focus has been on the paper, printing and wand technologies rather than on the data integration concepts. As a result, the linking mechanisms tend to be based on physical rather than logical resources. For example, a physical position within a document may link to a particular multimedia file. The result is that links are uni-directional and static. We however link at the level of logical resources, enabling links to be dynamic, context-dependent and multiple. In this way, we are able to achieve an integration of printed and digital information rather than simply of the underlying media. Further, users are able to freely author and share links, thereby enhancing the traditional qualities of paper as a static medium with dynamic linking features usually associated with digital media.

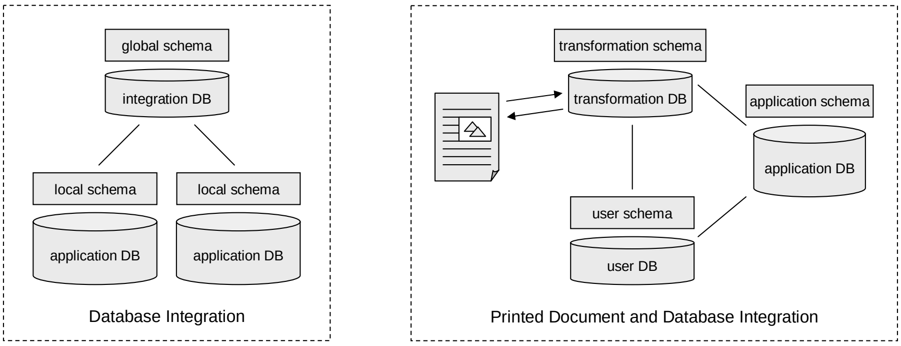
The key to semantic integration of printed and digital information lies primarily with the server rather than the client technologies. A suitable framework must be capable of dynamically mapping document positions to information objects and vice versa. It must manage not only the links, but also their semantics and have flexible means of classifying and associating both logical and physical resources. In addition, it must allow for managing information about both users and client devices. Finally, the framework must have a clear separation of content and presentation, enabling documents to be generated dynamically to suit client device, user and context. We have developed such an integration server in the context of a European project called Paper^{++} [21] which is developing and assessing innovative concepts and technologies aimed at enriching the use of paper in everyday settings. In line with many other current projects on data integration, we use the web as our basic integration platform.

In Sect. 2, we introduce our system and discuss how it relates to traditional data integration architectures. Section 3 then describes the general technologies for digitally augmented paper that underly our approach. In Sect. 4, we explain the overall architecture in terms of the relevant client and server components. Details of the XIMA and OMS data management components used to implement the integration framework are presented in Sect. 5 and 6, respectively. Finally, a discussion of related work is given in Sect. 7 and concluding remarks in Sect. 8.

2 Integrating Printed and Digital Data

As stated in the introduction, the details of particular data integration architectures depend on many factors such as the requirements of the target system and the characteristics of the data sources in terms of heterogeneity and autonomy of operation. Earlier projects focussed on the specific problem of database integration e.g. [9, 23]. More recent research has tended to address the more general problem of integration across many forms of information sources inclusive of databases, knowledge bases, digital libraries and web documents e.g. [28, 5]. While these architectures may vary greatly in terms of complexity, scope and approach, what they have in common is the use of some form of semantic integration component, be it a global schema, mediator or metadatabase.

For example, in Fig. 1.a, we show a simplified version of a typical database integration architecture where the local schemas are integrated into a global schema. Associated integration data may be stored at the global level to perform the necessary mappings between global semantic concepts and those of local application databases.



(a) (b)

Figure 1: Data integration architectures

When we want to integrate paper information sources with digital information sources, the situation is different in that there is no schema for a paper information source, i.e. there is no available metadata. The integration of information printed on paper and that of a database must be performed by linking documents, pages and parts of pages to information objects of the application database. As shown in Fig. 1.b, this is done through a transformation database which can map physical document positions to database objects and vice versa.

To obtain the required flexibility of being able to map physical document positions to multiple database objects and for that mapping to be dependent on both user and context, we have introduced the concepts of *shapes*, *layers* and *users* into our model. A shape describes a selectable area of a page. Generally, a digital wand (barcode reader) will return a coordinate position and the transformation database maps this to the corresponding shape, which is, in turn, associated with one or more objects of the application database.

Every shape is positioned on a specific layer. In the case that a point lies within the intersection of two or more shapes, the information object which is bound to the shape at the topmost layer of the set of active layers is returned. Furthermore, it is possible to selectively activate and deactivate specific layers which enables us to generate context dependent results by binding a particular position on a page to different resources according to the current active layer set.

More formally, assume that there are k layers numbered $1, 2, \ldots, k$, and that each shape S_i is associated with exactly one layer i.e. $\text{layer}(S_i) = j$ where $1 \leq j \leq k$.

For a given point (x, y), we write $(x, y) \in S_i$ to denote that the point (x, y) is contained within the shape S_i. If two shapes S_i and S_j are overlapping, i.e. there exists some point (x, y) such that $(x, y) \in S_i$ and $(x, y) \in S_j$, then the shapes must be on different layers i.e. $\text{layer}(S_i) \neq \text{layer}(S_j)$.

At any point in time, the set of active layers is a subset of all layers. Let A denote the set of all active layers. Then $A \subseteq \{1, 2, \ldots k\}$.

Then for a set of shapes $S = \{S_1, S_2, \ldots, S_n\}$, the selected shape associated with the given point (x, y) is given by the function s where

$$s(x, y) = S_m \text{ where layer}(S_m) = max\{j | S_i \in S, (x, y) \in S_i, \text{layer}(S_i) = j \text{ and } j \in A\}$$

Note that the layers should not be used to deploy personalised content to different users (e.g. by assigning a specific range of layers to every user and only activating the user specific layers). Rather, the system built-in user concept should be used to deploy user dependent content.

The user data component shown in Fig. 1.b manages information about the different users of the system. A user can define his preferences and can also be classified into different types of user groups. This enables an author of a digitally augmented document to, not only define different information anchors for different users, but also to bind different content to an information anchor based on the same user information. Therefore, we can use the same augmented paper to deploy different information based on a user's profile. A typical application of this user context-dependent information delivery is in an educational environment, where a teacher gets additional information for the same resource anchors. For example, a teacher may receive the solutions for exercises included in the course materials, whereas a student only obtains hints or links to other course material. The same user preferences can be used for different applications, i.e. a user does not have to specify new preferences for each application. For this reason, we show the user database in Fig. 1.b as a separate component.

Having introduced the basic idea of integrating paper and digital information sources, we now go on to discuss the details of technologies and architectures required to achieve this integration.

3 Digitally Augmented Paper

As described in the previous section, the integration of printed and digital information sources is based on the ability to detect page positions within a document. While the paper, printing and reader technologies required to fully support digitally augmented paper are still in their infancy, various forms of wand devices (e.g. barcode readers) are now emerging in the marketplace. Expected developments in the near future in terms of cheaper reading devices and means of invisibly encoding document positions on paper will surely have a dramatic effect on the spread and use of digitally augmented paper.

To annotate physical paper, we first need a technique to identify active areas of a sheet of paper. An active area is any shape defined by the author and might correspond to a paragraph, a text box, a single word, a figure or even part of a figure, etc. Further, active areas may be overlapping, for example, a specific word within a paragraph, text overlapping an image, or part of an image and the whole image. Ideally, a user should be able to point to an active area with a wand and trigger an associated action such as the display of a digital media file or a link menu, or the execution of some application system operation.

In the Paper^{++} prototype, detection of positions within a page is realised by printing on standard paper a grid of barcodes containing information about the page number and the corresponding position in terms of x and y coordinates using an inductive invisible ink. By pointing at the paper with a wand device which is capable of measuring differences of inductivity, we are able to read the barcodes and finally to decode the corresponding page number and the actual position of the reader.

We illustrate the basic idea of our system in Fig. 2. The shaded pattern across the page indicates the invisible barcodes that encode document positions in terms of coordinates. Logical elements within the page can be made selectable by defining corresponding logical shapes. The digital wand will return document positions and the server can then map these positions into corresponding shapes, which in turn are linked to digital information objects.

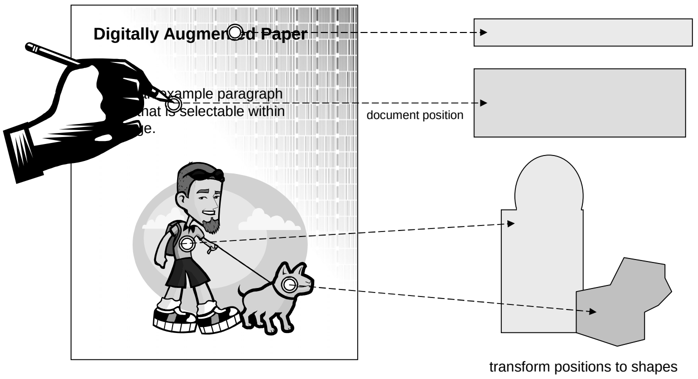


Figure 2: Mapping of page positions to defined shapes

For example, Fig. 2 shows a page containing a heading, a text paragraph and an image. The author has specified four selectable elements which are the heading, the text, the man within the image and the dog within the image. Each selectable element is specified by a shape as indicated on the right-hand side of the figure. Shapes may be of various forms such as rectangles, polygons or complex shapes such as the one shown for the man, which is composed from an ellipse and a rectangle. Each of these shapes is then associated to one or more objects of the application database. For example, the man's shape may be linked to a person object giving details of that person or possibly further digital multimedia resources such as images, videos or sound.

Note that, in our approach, we are not only capable of linking printed information to digital information, but also to dynamically link printed documents together. This is illustrated in Fig. 3. The left-hand side shows the page of the previous figure, which we can assume has an indirect link to an object for a particular breed of dog in a nature database. In the nature database, we assume that information is represented about various breeds of animals with associations to the countries in which these breeds can be found. On the right-hand side of the figure, we show an atlas and we further assume that someone has previously authored links from this atlas to the various countries represented in the nature database. In this way, links can be derived from one printed information source to another. For example, a user pointing to the dog in the paper document on the left, could be presented with various links to information, including a link to the atlas document and the position within the document where information about the country from which that breed of dog originates can be found. It is important to realise that here the link from the image of the dog to the map within the atlas has not been explicitly specified by an author. Rather it is dynamically derived based on authored links to concepts of the nature database and the associations within this database.

With the notion of digitally augmented paper that we have presented, it is possible for users to browse back and forth between printed and digital information and also between different sources of printed information. This may contribute to avoiding problems of mode-locking, where a user switching from paper to digital information locks into the digital world and is distracted from the original document. It could be said that our approach not only allows the digital annotation of physical objects, but also the "physical annotation" of digital objects. Even without any additional multimedia content, our system will provide a powerful tool allowing users to classify and build links between physical documents.

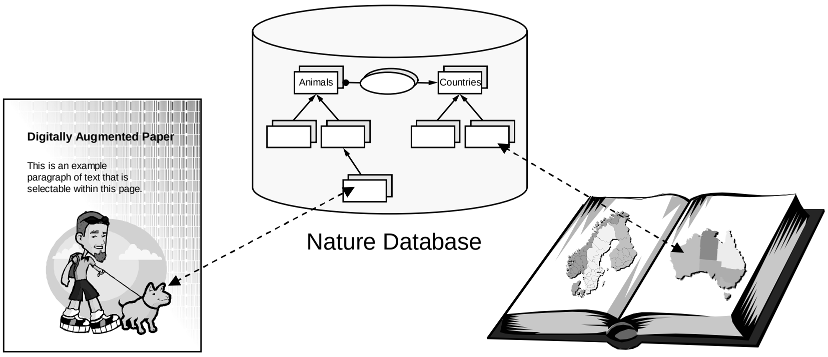
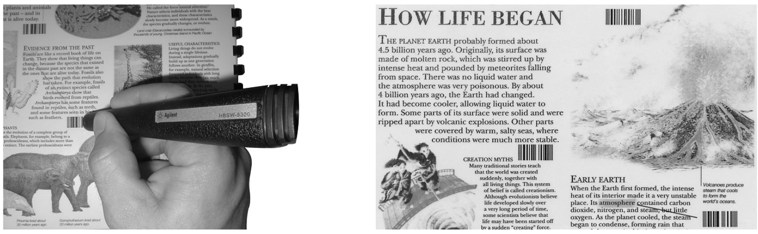


Figure 3: Linking paper documents

An important issue is how best to provide user with feedback concerning the location of active areas within a document and also the, possibly context-dependent, associated actions. Whereas the location of the information anchors is obvious in the case of visible barcodes (see Fig. 4), a mechanism will be required to inform users of the presence of active areas when using invisible barcodes. In an authored document, this can be done by highlighting those parts of a page containing links to digital information (as indicated in Fig. 4.b with the highlighting of the word that is the active area corresponding to the barcode in the lower right corner). For specific applications, it may even make sense to omit any visible hints about additional information and leave the task of finding the links completely to the user (e.g. in a children's book where it may be part of a learning task to find additional information such as realised in the LeapPad suite of books [15]). In the case of user generated links, the simplest way to visualise the information anchors is to combine the wand device with a marker pen allowing the user to highlight areas on paper while adding a new link. Further, we have experimented with the use of sound as a possible feedback mechanism.



HOW LIFE BEGAN

THE PLANET EARTH probably formed about 4.5 billion years ago. Originally, its surface was made of molten rock, which was stirred up by intense heat and pounded by meteorites falling from space. There was no liquid water and the atmosphere was very poisonous. By about 4 billion years ago, the Earth had changed. It had become cooler, allowing liquid water to form. Some parts of its surface were solid and were ripped apart by volcanic explosions. Other parts were covered by warm, salty seas, where conditions were much more stable.

CREATION MYTHS

Many traditional stories teach that the world was created suddenly, together with all living things. This system of belief is called creationism. Although evolutionists believe life developed slowly over a very long period of time, some scientists believe that life may have been started off by a sudden "creating" force.

EARLY EARTH

When the Earth first formed, the intense heat of its interior made it a very unstable place. Its atmosphere contained carbon dioxide, nitrogen, and steam, but little oxygen. As the planet cooled, the steam began to condense, forming rain that

Volcanoes produce steam that cools to form the world's oceans.

(a) (b)

Figure 4: Annotated nature encyclopedia page

The development of printing technologies, barcode schemes and wand devices on which such systems could be based is very much an active area of current research. Solutions do exist, but at present they tend to be expensive and based on visible encoding patterns. However, it is expected that cost-effective solutions which are almost invisible and with barcode schemes that provide a good level of position granularity will be available in the near future. Therefore an important factor in the development of our server was to ensure that it is independent of any particular technologies and encoding schemes. What we assume is simply the delivery of a document position. In our discussion of related work in Sect. 7, we describe some alternative technologies to those presented above.

4 Paper^{++} Server

Our framework is based on a classic client server architecture. On the client side, we have the wand device enabling us to read the document positions using inductive barcodes printed on digitally augmented documents. The client device transmits all of its data to a server which then does further processing of the information. The Paper^{++} server consists of three main components which are shown in Fig. 5. The barcode decoder and visualisation component is responsible for both decoding the barcodes received from the input device and then later presenting the results returned by the eXtensible Information Management Architecture (XIMA) component after processing the request. All data is managed by the interaction and application component which is based on the OMS Java data management system [13, 12].

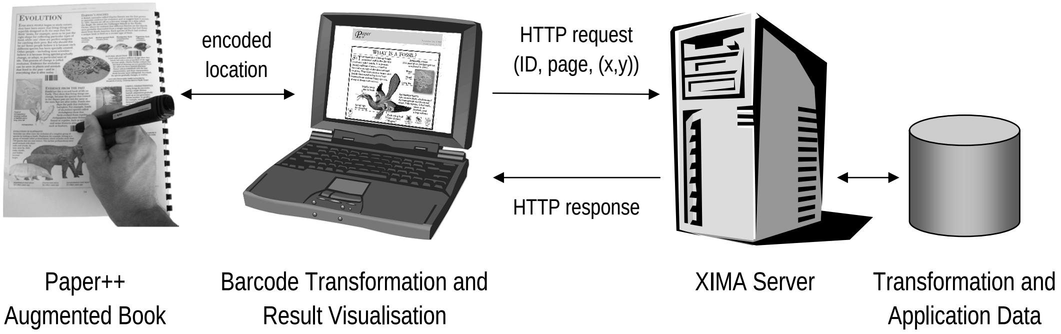


Figure 5: Paper^{++} architecture overview

In a typical Paper^{++} user session, the pointing device transmits recognised barcode information in a preprocessed ASCII format to the barcode transformer, which then decodes relevant information such as the page number and the x and y positions from the barcode. In the next step, an HTTP request including this information about the wand device's current position on the paper document is sent to the XIMA server component. Based on the interaction model and its transformation data, the server fetches the corresponding information from the application database and returns the result in the preferred format of the digital presentation device. For example, a regular web browser would receive an HTML formatted result page, while a WAP-enabled phone would receive a WML result page for the same request.

In Fig. 5, the digital presentation device is a laptop computer which also serves as the barcode transformation component. However note that, in theory, it is not necessary that the barcode transformation and result visualisation take place on the same device. This means that in a public space, for example, the result could be displayed on a large global display screen, while the transformations take place on users' individual portable devices.

It is important to note that all operations on the server side are independent of a particular paper, printing or digital wand technology. Our initial prototype application [7] has been developed using existing barcode reader technologies, but the barcode transformation process can easily be adapted to any form of device capable of detecting the position on a sheet of paper. All other components are completely independent of the form of input device, although certain technologies may be able to support extended functionalities in terms of user interaction. In Fig. 6.a, we present the configuration that has been used for our initial user studies — the nature encyclopedia application running on a laptop computer and an off-the-shelf barcode reader to handle input processing. The same application running on a Pocket PC with a 802.11b WLAN card is shown in Fig. 6.b.

(a) (b)

Figure 6: Nature encyclopedia application running on a laptop and on a Pocket PC

To remain flexible in handling user input, we defined the entire logic of the input device in separate interfaces. Through a simple replacement of the concrete Java class implementing the wand device interface, we can therefore easily switch to different input devices.

The XIMA server and the OMS Java system are general components for access to and the management of data. In the case of the Paper^{++} server, they not only manage all the transformation, user and application data, but also perform the necessary request processing inclusive of mapping document positions to application database objects and generating client-specific presentations of data. In the following sections, we describe first the XIMA framework, which can be considered as a universal web-based access layer for OMS Java, and then follow this with some comments about the OMS Java data management system in general and the Paper^{++} server databases in particular.

5 XIMA Framework

The eXtensible Information Management Architecture (XIMA) was developed to support universal client access to information systems based on web technologies [24]. Many existing tools for web site engineering either rely on a specific form of client such as an HTML browser, or they require that different forms of clients be handled separately meaning that it requires significant development time to port an application to another form of client device. We wanted a universal client framework that required minimal development effort to access web documents via another form of client device. Any additional effort would rather go into optionally customising the presentation to suit the device in question. Such general frameworks are particularly important when one considers how dynamic the world of mobile devices currently is.

An overview of the XIMA server component is given in Fig. 7. For a specific application, all client access is via a single Java servlet — the Entry Servlet. This means that only a single URL need be known for a specific web site, rather than having different URLs for different types of clients. The Entry Servlet detects the user agent type from the HTTP request header and delegates the handling of the request to the appropriate servlet. For example, we show in Fig. 7 servlets to handle requests from HTML browsers, XML browsers and also WAP phones in terms of WML browsers. In addition, we support access via Imode mobile phones and regular speech telephones based on the speech recognition and synthesis support provided by VoiceXML servers.

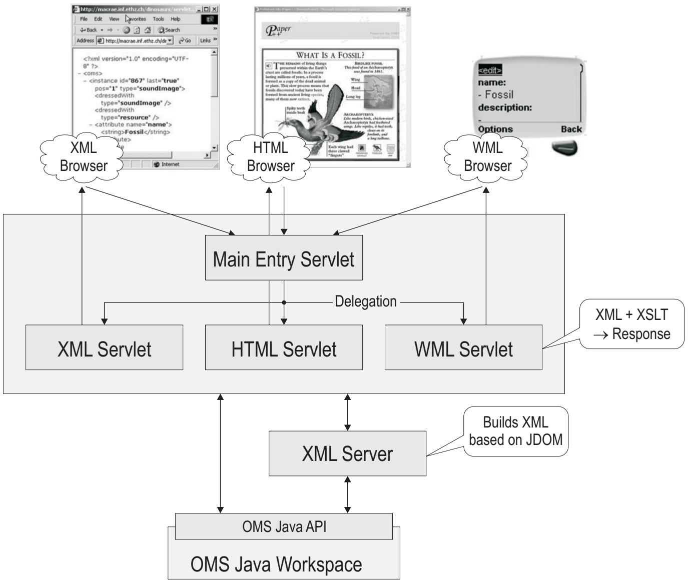


Figure 7: XIMA server component

The request handling servlets then access the database by connecting to an OMS Java workspace via the OMS Java API. The connection may either be direct or via the XML server component. Direct connections deal with requests that do not involve data retrieval such as checking the membership of an object in a collection or performing updates. Any requests that involve the retrieval of data go through the XML server which forwards them to the OMS Java data management system and generates XML representations for any data objects to be returned to the servlets. The requesting servlets then use the appropriate XSLT templates to transform the XML results to the corresponding responses to be returned to the client.

A point to note in this general architecture is that we are not storing any XML documents, but rather generating them dynamically from the application data which is stored according to the information model. Since the information model is a loosely-coupled graph model based on object collections and associations, this gives much more flexibility than the rather restrictive hierarchical-based models imposed by XML structures. At access time, a particular hierarchical view on data is derived and the appropriate XML structure generated.

Using generic XSLT templates for the various client devices, we are able to provide generic browsers and editors for the current set of client device types. Adding a new type of client device involves implementing the corresponding servlet and writing the appropriate XSLT templates. Specific application interfaces are supported through the customisation of XSLT templates and the specification of document structures in terms of how document content is built dynamically from one or more individual information objects. It is beyond the scope of this paper to describe all the details of the XIMA framework, but further information can be found in [24].

6 OMS Data Management Component

When discussing the general integration architecture in Sect. 2, we showed three database components — the transformation database, the user database and the application database. In practice, these may all be within a single database or they could be separate databases.

For example, the user preferences may be general and apply to more than one application and, hence, we could choose to store them in a separate database and would use this in conjunction with a number of application database components. If the application database exists already, then it makes sense to perform the integration through a separate transformation database — especially if the application actually involves a number of application databases (or even other forms of digital information source). In the actual prototype systems that we have built so far for Paper^{++}, the databases were created from scratch and hence we integrated the components within a single database. However, it would be no problem to distribute the components and we even have a version of OMS that specifically supports peer-to-peer connection among distributed database components [19].

The transformation, user and application data is stored using the OMS Java data management framework which is based on the OM object model [18]. OM is a generic, object-oriented model that supports information modelling through a two-level structure of classification and typing. Entities are represented by objects with attributes, methods and triggers as defined by the associated object types. The classification of entities is done through semantic groupings represented by named collections with a specified member type. The member type of a collection not only constrains its membership, but also specifies a contextual view of objects accessed through that collection. Thus, we can have role-dependent object views and behaviours by associating different types with collections according to the roles that they represent.

OM has a high level association construct which enables relationships to be classified and manipulated directly. The association construct further supports the decoupling of information objects since objects tend to be referenced through the association concept, rather than through object reference attributes. Last but not least, the model includes a powerful set of operations over objects, collections and associations.

The expressive features of the OM model enable us to capture the semantics of application domains in terms of a simple, but powerful set of constructs. Its support for the direct representation and manipulation of associations is particularly useful in supporting link management in systems that support hypermedia browsing such as Paper^{++}.

In Fig. 8, we show part of the application database model for a nature database that was developed for a Paper^{++} prototype to integrate printed and digital information associated with a nature encyclopaedia. As can be seen in the model, all objects of the application model are *resources*. This means that they can be associated with anchors defined in printed documents. Using the *relatedTo* association, each resource can further be linked to other resources. The system still allows links to be built to simple multimedia objects such as pieces of text, images or sound clips as shown in the upper part of the figure. However, it becomes more interesting if we start linking parts of a physical document to so-called *application objects* representing concepts of a specific application domain.

In the case of the nature database, we have the concepts of species, habitats, different classifications, locations etc. Further, these concepts are connected by specific associations (e.g. *hasClassification*, *hasHabitat*). In this way, we can express the fact that a species is found at a specific location, and we can give each species a classification and add other domain specific semantics. Let us now assume that a user is reading the printed version of the nature encyclopaedia and he wants to get more information about a particular species. By following the link to the digital media, he gets additional information about the animal and can browse the digital information. However, as described earlier in the paper, he can also follow links back to the paper world such as an atlas giving information about the country in which a species is found.

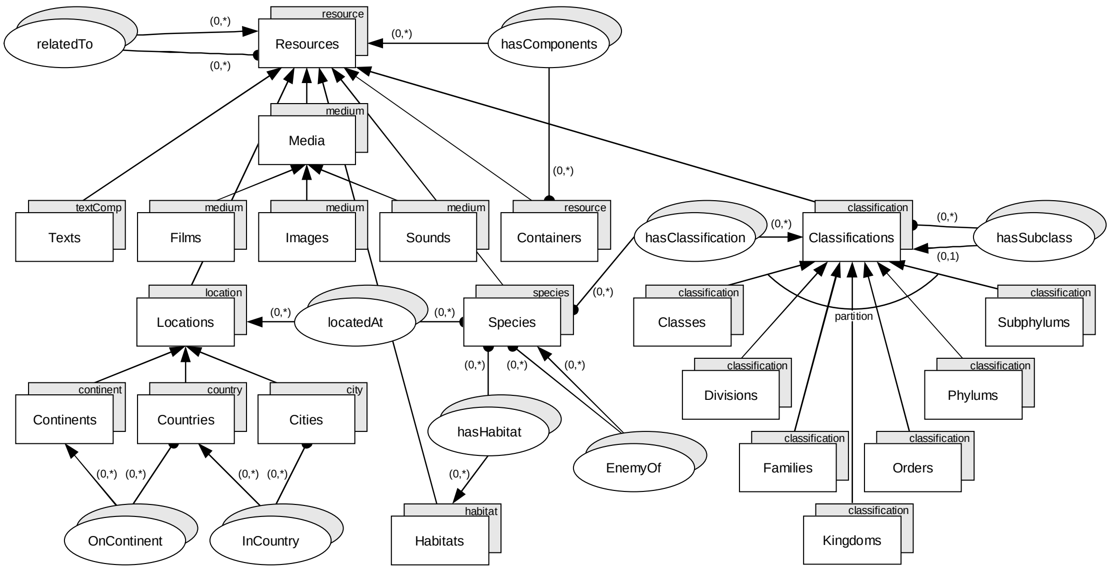


Figure 8: Nature encyclopaedia application model

To bridge physical paper and the digital medium, the transformation model defines how a document's meta information is represented. As shown in Fig. 9, a document can contain multiple pages and each page can contain multiple resource anchors (active areas). The actual information anchors are defined by shapes such as rectangles, circles or polygons. The transformation data is used to check whether, for a specific position on a document's page, there exists one or more information anchors, i.e. it allows the system to test if a point lies within one or more shapes. Finally, every resource anchor is associated with an information resource of the application database. We have to point out that such an association is always bidirectional, which means that later we can, not only fetch the information associated with a specific shape, but also get the locations of all active areas of printed documents associated with a digital object stored in the application database.

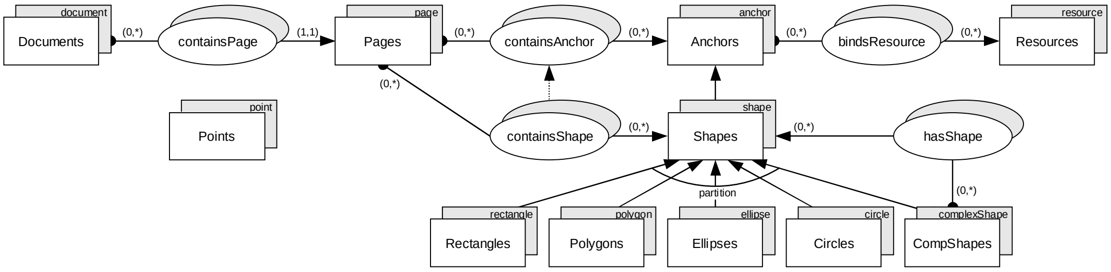


Figure 9: Transformation database component

Although not shown in Fig. 9, the model also associates shapes with layers. This enables overlapping shapes and the possibility to link particular parts of a shape to different resources. In the case that a position lies in more than one shape, the one on the topmost, active layer is returned as described in Sect. 2. It would be possible for example to use this to provide a form of "zoom in" by deactivating the topmost layer each time the user points to a document element.

Last but not least, the presentation data model contains information about supported types of client devices and templates for the final rendering of generic XML results. Based on a dynamically generated XML representation of the objects stored in the OMS Java system, the client specific rendering is realised using the Extensible Stylesheet Language Transformation (XSLT). This allows us to not only support a variety of client devices, but also to enable variations in visualisation for the same type of device. A regular user of the application can customise the user interface by defining his own stylesheets, whereas an application can already provide different visualisations for specific groups of users defined by the user model.

7 Related Work

As we will describe in this section, there exist numerous different approaches to encode information on a piece of paper. However, we believe that the power of digitally augmented paper strongly depends on the functionality of the server component handling the requests and not only on the client technologies. Users should be able to freely browse back and forth between the paper and digital worlds, dynamically creating not only annotations, but also links between paper and digital resources, or even between paper resources. In this way, paper would become a *first-class citizen* within hypermedia systems. The key to achieving this vision lies in using powerful database technologies on the server side that not only provide flexible and powerful dynamic link management, but also manage application semantics. To explain further what we mean by this, we describe the linking processes supported by existing projects and their limitations.

We start by comparing our system with the *Intelligent Paper* approach described in [6]. They propose a similar solution where standard sheets of paper are entirely covered with invisible printed marks. In terms of the position detection mechanism, the two approaches are quite similar, although there are some differences in how information is encoded within the barcodes. Instead of using globally unique identifiers on every page, we separate the document identifier from the page identifier and only encode the page identifier within each page of a specific document. However, the major difference between our approach and theirs is in how the mapping from the physical location on paper to the digital content is managed. The *Intelligent Paper* project proposes to use the Adobe Acrobat suite of products to build an electronic counterpart of the paper document with embedded links for every physical page. In contrast, our solution retains the two separate paper and digital worlds and has no need of an equivalent digital version of the physical page. Rather it creates logical links between the two media allowing elements on a physical page to be mapped to the appropriate digital objects.

Using barcodes to encode information on a piece of paper is just one possible solution. Another interesting approach from Xerox is based on DataGlyphs [8] to store data on physical paper. They use special cover sheet forms containing DataGlyphs to build paper user interfaces [11]. The cover sheet is placed in front of a document in a scanner or a fax machine connected to a document services system and allows the definition of specific actions to be taken on the document which follows it. However, the augmentation is not on the document itself rather than on an additional piece of paper.

Another technology developed by a Swedish company called Anoto [1] uses a special pattern of “invisible” dots to print a grid over a page. The corresponding digital pen uses a digital camera to track movement. However, the main focus of the Anoto C-Pen and other electronic pens [10, 20] is to digitise information and transmit it to a client device, whereas our goal is to digitally augment the physical medium.

An alternative possibility to create information anchors on physical paper is described in [2] where a highlighter pen is augmented with a small camera. Simple computer vision and pattern recognition techniques allow a user to make marks on paper. However, the stored patterns are not associated with any document meta information which makes it difficult to support bidirectional links enabling a user to browse back and forth between physical paper and digital media.

In [17, 27, 16], a video camera is mounted above an office desk pointing down at the work surface. Additionally a projector is used to display objects on the table. A drawback of these solutions is that the paper always has to be processed on a digital desk and therefore we lose one of the main benefits that paper affords in terms of its mobility. Existing applications which are based on the digital desktop metaphor (e.g. an augmented version of *Alice's Adventures in Wonderland* [4]) do not allow browsing from the digital to the physical medium.

Instead of using a visual annotation to identify a document, Radio Frequency Identification (RFID) tags and the corresponding readers can be used as described in [26]. The RFID tags enable documents or individual pages to be identified: They do not allow specific parts of a page to be addressed. A combination of technologies is presented in [3] where electronic field sensors located in the book binding sense the proximity of the reader's hands and control audio parameters, while RFID tags embedded in each page allow page identification. As described earlier, within the Paper^{++} project, we separate the document identifier from the page identifier. Whereas the identification of pages and elements within a single page is done by using the concept of invisible barcode grids, the idea is to use RFID tags to identify whole documents (e.g. a single book).

Numerous existing projects, e.g. [2, 25, 14], only allow links to single objects such as a piece of electronic text, a sound or a movie clip. They have no support for following links to other related digital resources or back to other paper resources. Thus, the focus of these projects is the ability to annotate a simple sheet of paper with multimedia content. On the server side there is a simple database (in reality probably a simple file), that maps positions (or even single barcodes) to specific digital resources. The processing of a request consists merely of displaying the digital resource.

8 Conclusions

We have described an approach to digitally augmented paper that enables paper to be fully incorporated into digital information spaces as a browsing device. In contrast to existing projects for digitally augmented paper, our approach is based on dynamic links between logical elements of the information sources rather than on hard-coded links between physical media. The key to achieving this was to use a database approach and exploit database concepts in terms of information abstractions, metadata, dynamic links and document generation.

The integration platform was implemented using the existing XIMA object-oriented data management framework, which supports universal client access. In the context of the Paper^{++} project, we have developed two applications — one is the nature encyclopaedia and the other deals with information about the art world. These prototype systems are based on conventional barcode technologies, which, although not ideal, have at least been sufficient to test our models and also to carry out initial user studies. Since the server was developed with generality in mind, it will be a simple matter to change to new paper, printing and wand technologies as they emerge in a reliable and cost-effective form.

Links are bidirectional and enable users to move freely back and forth between paper and digital media. This means that we can combine the features of paper (static, excellent readability properties, flexible, inexpensive, ubiquitous) with those of a digital medium (dynamic, can store a lot of information) and, using current skills, have optional access to new functionality. In addition to general link authoring tools, we are currently developing an automatic link generator component that allows us to build dynamic links in addition to the explicitly authored ones. The link generation is based on user monitoring and tries to detect trails "walked" by numerous users.

Acknowledgements

The work reported in this paper is part of the European project Paper^{++}, under the Disappearing Computer Programme (IST-2000-26130). We thank the other members of the project for their contributions and feedback on our work.

References

[1] Anoto AB, http://www.anoto.com.

[2] T. Arai, D. Aust, and S. Hudson. PaperLink: A Technique for Hyperlinking from Real Paper to Electronic Content. In *Proceedings of ACM CHI 97 Conference on Human Factors in Computing Systems*, volume 1 of *Papers About Paper*, 1997.

[3] M. Back, J. Cohen, R. Gold, S. Harrison, and S. Minneman. Listen Reader: An Electronically Augmented Paper-Based Book. In *Proceedings of ACM CHI 2001 Conference on Human Factors in Computing Systems*, Designed Experiences/Experienced Designs, 2001.

[4] H. Brown, R. Harding, S. Lay, and P. Robinson. Active Alice: Using Real Paper to Interact with Electronic Text. *Lecture Notes in Computer Science*, 1998.

[5] S. Chawathe, H. Garcia-Molina, J. Hammer, K. Ireland, Y. Papakonstantinou, J. Ullman, and J. Widom. The TSIMMIS Project: Integration of Heterogeneous Information Sources. In *16th Meeting of the Information Processing Society of Japan*, 1994.

[6] M. Dymetman and M. Copperman. Intelligent Paper. *Lecture Notes in Computer Science*, 1998.

[7] D. Frohlich, E. Tallyn, N. Linketscher, B. Signer, G. Adams, and P. Luff. Paper^{++} Deliverable 8, Initial Paper^{++} Assessment. Technical report, April 2002.

[8] D. L. Hecht. Printed Embedded Data Graphical User Interfaces. *IEEE Computer*, March 2001.

[9] D. Heimbigner and D. McLeod. A Federated Architecture for Information Management. *ACM Transactions on Office Information Systems*, pages 253–278, 1985.

[10] InMotion, e-pen, http://www.e-pen.com.

[11] W. Johnson, H. Jellinek, L. Klotz, Jr., R. Rao, and S. Card. Bridging the Paper and Electronic Worlds: The Paper User Interface. In *Proceedings of ACM INTERCHI'93 Conference on Human Factors in Computing Systems*, Finding and Keeping Information, 1993.

[12] A. Kobler and M. C. Norrie. OMS Java: Lessons Learned from Building a Multi-Tier Object Management Framework. In *Proceedings of the Workshop on Java and Databases: Persistence Options*, Denver, USA, January 1999.

[13] A. Kobler, M. C. Norrie, B. Signer, and M. Grossniklaus. OMS Java: Providing Information, Storage and Access Abstractions in an Object-Oriented Framework. In *Proceedings of the 7th International Conference on Object-Oriented Information Systems (OOIS'01)*, Calgary, Canada, August 2001.

[14] B. M. Lange, M. A. Jones, and J. L. Meyers. Insight Lab: An Immersive Team Environment Linking Paper, Displays, and Data. In *Proceedings of ACM CHI 98 Conference on Human Factors in Computing Systems*, volume 1 of *Computer Augmented Environments*, 1998.

[15] LeapPad Learning System, LeapFrog Enterprises, Inc., http://www.leapfrog.com.

[16] W. E. Mackay. Augmented Reality: Linking real and virtual worlds. In *Proceedings of ACM AVI '98, Conference on Advanced Visual Interfaces*, 1998.

[17] W. Newman and P. Wellner. A Desk Supporting Computer-Based Interaction with Paper Documents. In *Proceedings of ACM CHI'92 Conference on Human Factors in Computing Systems*, Desks, Video, and Screens, 1992.

[18] M. C. Norrie, , A. Steiner, A. Würgler, and M. Wunderli. A Model for Classification Structures with Evolution Control. In *Proceedings of 15th International Conference on Conceptual Modelling (ER'96)*, Cottbus, Germany, 1996.

[19] M. C. Norrie, A. Palinginis, and A. Würgler. OMS Connect: Supporting Multidatabase and Mobile Working through Database Connectivity. In *Proceedings of Conference on Cooperative Information Systems*, New York, USA, 1998.

[20] OTM Technologies, VPen, http://www.otmtech.com.

[21] Paper^{++}, IST-2000-26130, http://www.paperplusplus.net.

[22] A. J. Sellen and R. Harper. *The Myth of the Paperless Office*. MIT Press, 2001.

[23] A. Sheth and J. Larson. Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases. *ACM Computing Surveys*, pages 183–236, September 1990.

[24] B. Signer, M. Grossniklaus, and M. C. Norrie. Java Framework for Database-Centric Web Engineering. In *Proceedings of the 4th Workshop on Web Engineering (in conjunction with 10th International World Wide Web Conference)*, Hong Kong, Mai 2001.

[25] L. J. Stifelman. Augmenting Real-World Objects: A Paper-Based Audio Notebook. In *Proceedings of ACM CHI 96 Conference on Human Factors in Computing Systems*, volume 2 of *Scribbling, Sketching, Drawing, and Writing*, 1996.

[26] R. Want, K. P. Fishkin, A. Gujar, and B. L. Harrison. Bridging Physical and Virtual Worlds with Electronic Tags. In *Proceedings of ACM CHI 99 Conference on Human Factors in Computing Systems*, volume 1 of *Tagging and Tracking Objects in Physical UIs*, 1999.

[27] P. Wellner. Interacting with Paper on the DigitalDesk. *Communications of the ACM*, 36(7), July 1993.

[28] G. Wiederhold. Mediators in the Architecture of Future Information Systems. *IEEE Computer*, pages 38–49, 1992.