Information Server for Highly-Connected Cross-Media Publishing

Moira C. Norrie[a], Beat Signer[a]

[a]Institute for Information Systems, ETH Zurich, CH-8092 Zurich, Switzerland

The final publication is available at http://dx.doi.org/10.1016/j.is.2004.11.003

Over the last decade, we have seen a significant increase in the number of projects aiming for integration of different kinds of media (mixed-media integration). However, most existing approaches tend to focus on the media technologies rather than on concepts for information integration and linking that enable users to move freely back and forth between various media information sources. In this paper, we discuss the issues of information semantics and granularity that arise in the design of highly interactive mixed-media information systems and present a general, flexible information server that meets the requirements of publishing information on different output channels (cross-media publishing). Specifically, we introduce the iServer framework as a generic link management and extensible integration platform and digitally augmented paper is presented as one specific application of the iServer technology. A case study shows how cross-media publishers could profit from using more elaborate information systems and some of the authoring issues of mixed-media information environments are discussed.

1. Introduction

In recent years, there have been significant developments within the hypertext community in the area of open hypermedia systems that enable the dynamic linking of a variety of media information sources [1–3]. However, most of the projects realised to date tend to focus on the linking of rather simple types of digital media such as textual information, image files, video files and other digital multimedia content. In this context, much less effort has been placed on the capability to link physical and digital resources, enabling, for example, the integration of printed and digital information.

In contrast, several projects in the realm of pervasive computing have tried to integrate real and virtual environments by linking together physical and digital artefacts (mixed-media integration) [4–6]. Nevertheless, in this case, most of the approaches focus on hardware issues such as the detection and tracking of physical objects and utilise very simple integration models that do nothing more than link entire physical and digital resources together. By neglecting aspects of information semantics and performing linking at such a coarse level of granularity, the level of connectivity tends to be minimal and, in many cases, linking is restricted to a single direction and a single level. Thus, while it would be possible to activate a link from a physical object to an image, it would not be possible to follow further links from the image to other digital or physical resources.

What we want to show is that by approaching the problem from an information point of view we can take the integration of the physical and digital worlds one step further and realise highly-connected systems that can link together elements of resources and enable users to move freely back and forth between physical and digital information sources. Further, by abstracting the core concepts of mixed-media links, we were able to build a generic link management framework, which can easily be extended to support any kind of digital or physical media type. In this way, we have been able to achieve the flexibility of open hypermedia systems while bridging physical and virtual worlds.

The integration of printed and digital resources is just one example where we can get much richer information environments by combining two different media instead of trying to replace paper completely with electronic documents. This enables both the producers and consumers of information to benefit from the advantages that a particular medium offers in certain situations. For example, while electronic documents are easier to distribute and update, many users prefer to read and annotate paper documents. Further discussions on the affordances of paper and reasons for its continued use can be found in [7].

A range of technologies are emerging that enable active links to be encoded on paper in such a way that special reader devices can detect and activate these links by means of sending requests to a server which then processes the requests and produces the appropriate responses. Generally, we classify such technologies as supporting *digitally augmented paper* and would recommend the following references to readers interested in details of the various technologies available [8–11].

The question remains as to the potential uses and models of interaction afforded by digitally augmented paper. How can we fully exploit the potential of paper as a client device and integrate it seamlessly into mixed-media systems? While a number of research projects have proposed different variations of digitally augmented paper and presented their visions for its use, none have really addressed the issue of making paper a *first-class medium* in the context of hypermedia systems. It should not only be possible to link from paper to digital information, but also to physically augment digital content or even to enable links from paper to paper.

In this article, we describe how we support digitally augmented paper as a specific application of our more general mixed-media integration server (iServer). A case study is used to motivate the need for such servers and their requirements. Further, we show how the important issues of information semantics and granularity impact on the level of connectivity of the resulting information environment. Finally, we discuss the issues that cross-media publishing raises for publishers and the forms of authoring tools that could support the publishing activity.

We begin in Sect. 2 by discussing the requirements of mixed-media integration and the status of existing solutions. In Sect. 3 we introduce our mixed-media integration server (iServer) which was designed based on these requirements and manages link information on a fairly abstract level. Digitally augmented paper is then introduced as a specific application of the iServer technology in Sect. 4, showing the steps involved in supporting new types of media in the general link management framework. Having discussed the basic link issues, we then show in Sect. 5 how application databases can be integrated to enrich mixed-media information spaces and support higher levels of connectivity. In addition, we explain how user modelling supports customised information deployment. Section 6 discusses some general issues of the authoring of mixed-media material and presents our authoring tool for digitally augmented paper. Concluding remarks are given in Sect. 7.

2. Mixed-Media Integration

In this section, we begin by motivating the need for mixed-media integration and then go on to outline some of the drawbacks of existing approaches. We conclude the section by elaborating on some of the requirements to build more flexible and highly-connected mixed-media information systems.

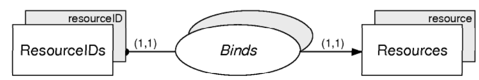
It is common nowadays for publishers to produce a package of related materials on different media. For example, the BBC (British Broadcasting Corporation) often produces books and also websites to accompany television documentary series. As a specific example, for their series on ocean life called *Blue Planet*, they have an associated book and a website with games, quizzes and a fact file on different species [12]. In addition, this series was adopted by the Open University [13] in a course on oceanography and a course text book was published to be used in conjunction with the TV programmes which are also available on video and DVD.

How is linking across media currently done? As a first remark, it is important to note that, in many cases, it is not done at all. What might at first appear to be a package of related materials, is often actually a set of quite independently produced media. For example, in our initial experiments, we used a printed book and a CD-ROM of a children's nature encyclopaedia and studies of the material clearly showed that, not only were they marketed as separate products, but also the content was quite different.

In the case of the Blue Planet material, the Open University course book has been designed so there is a close correspondence with the TV material and students can supplement learning by watching clips from the series. At the end of a section of text, there is a recommendation to view all or part of a programme. These recommendations are indicated by a video cassette symbol placed alongside the text as shown in Fig. 1. In the notes section shown at the bottom of the page, timing information is given for video segments along with a text that either describes what is seen in that part of the video or provides supplementary information. In some cases, the notes may reference other sections of the course text where something mentioned briefly in the video clip is explained in detail.

Currently, the linking between the printed course text book and the video material is entirely manual in that readers are simply given printed signals to indicate when they should manually activate another media form. Our aim would be to automate this process by allowing users to activate the video through a simple pointing or swiping process on paper. A simple means of doing this would be to use a special printed encoding of a unique identifier (ID) on the paper that could be detected by a special reader. For example, a barcode could be printed on the page where there is currently a video symbol and a standard barcode reader used to activate the link. A simple server would then map the unique ID to the appropriate file name and play the corresponding video. Alternatively, the company Anoto [14] has developed a pattern of tiny visible dots for encoding information on printed documents and this can be used in conjunction with special reader devices such as the MyPen [15], the Logitech io Personal Digital Pen [16] or the Nokia Digital Pen [17] to detect encoded information.

Most of the existing proposals for creating links between physical paper and digital information are based on exactly this concept of printing unique identifiers on the paper document. These unique digital resource IDs are then used to fetch corresponding multimedia resources based on a look up in a fairly simple mapping table [18–20]. In Fig. 2, we present such a basic mapping table together with the related model which consists of an association `Binds` between a set of `ResourceIDs` and a set of `Resources`.



Resource ID	Resource
01237685	angler.jpg
01426782	blacksmith.jpg
04632534	bonefish.mov
18735676	bonito.gif
27898891	cabezon.jpg
45372679	cod.mov
...	...

Figure 2. Simple mapping of IDs to multimedia resources

What comments can we make to such a simple linking scheme? The first is that the links are unidirectional, only supporting links from the printed document to digital resources. Given a particular digital resource such as a video, there would be no way to find the various places within a paper document, or set of documents, that link to that video. Secondly, the activation of a link causes a file to be displayed using the appropriate application. However, there is no way then to link to further resources within the link architecture itself. For example, after following a link to a video, we might want to have links available to related video clips or supplementary texts (printed or digital) about the clip. Since the proposed simple linking scheme does not support links between different resources, the potential hypermedia features of the linked resources themselves have to be used for this purpose. For example, if the digital resource is an HTML page, then HTML specified links within the page will provide additional linking possibilities. We therefore say that the link paths of this simple linking approach are restricted to length 1, whereas hypermedia systems support link paths of arbitrary length.

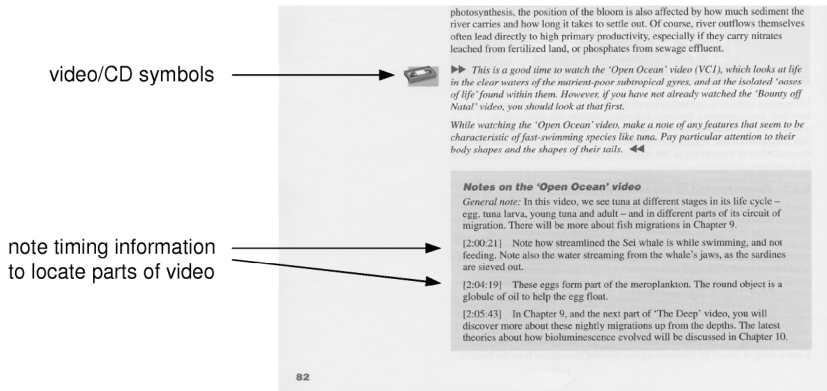


Figure 1. Page of Open University course book

Such a solution would therefore not even support the various forms of manual linking that are supported by the existing end of section notes in the Blue Planet course book. These limitations are in many ways due to the over-simplistic information model which simply binds IDs to files. To implement more interactive information environments, a first requirement is to support *bidirectional links* so that we can not only follow a link from its physical anchor to the related digital resource, but additionally always have the possibility to activate a link back from the digital to the physical media. To enable those links from the digital world back to physical paper, it is no longer sufficient to have unique identifiers as anchors in the physical media. In addition, we also have to manage supplementary information about the relative position of a link anchor and/or link target within a printed document. The relative position could be specified in terms of the document ID (possibly name), the page ID (number) and some spatial information about where the link anchor or target is located within the page. Based on this metainformation, the system could provide links automatically from the digital to the physical media by signalling to a user which document should be used and on which page of the corresponding document the link can be found. This would enable the forms of video-to-book references currently supported in the notes sections of the Blue Planet course text where the entry associated with a video sequence references another section of the book. For example, at the end of section 3.3, the notes on the 'Tidal Seas' video include:

> *[0:56:54] The influence of hurricanes on sea-level will be explained in Section 3.6.*

The inclusion of timing information for video sequences in the notes section together with descriptions of the contents of these sequences shows that the publishers want to provide links at a finer level of granularity than entire sections of text and video programmes. Ideally, we would like to be able to link elements of media resources and not just entire resources. Thus, we would like to be able to link words, phrases or images within a section to video sequences and vice versa. This would enable the publisher to provide a much more highly-connected, and hence highly-interactive, information environment.

To provide even richer information environments, it would be useful to have metadata about the semantic content of the elements being linked. Instead of simply linking resources such as video, text etc. together, resources could consist of data or metadata objects of a database that describe a particular entity or concept of the application domain. For example, chapter 7 of the Blue Planet course book is about coral reefs and section 7.2 describes where coral reefs can be found.

> *"They are widespread throughout the tropical Indian Ocean, in the western*

> *tropical Pacific, in the Carribean and along the coast of southern Brazil, but are found only in restricted areas in the eastern Pacific and the eastern Atlantic."*

It would be useful to provide links from the various geographical regions mentioned to maps showing these regions. By linking to an image resource, the appropriate map could be displayed when the link is activated. However, if we had a database with information about geographical regions, then we could link to an object with information about that region, inclusive of a map, and therefore provide the option of getting further information about that region and also the possibility of finding out about associated regions such as sub-regions or adjacent regions. If the database also contained information about animal and plant life, then it would be possible to follow links to information about other species found in these locations. In this way, application databases can be used to enrich the information environment, providing a valuable information source, inclusive of semantic associations that provide a means of automatically generating bi-directional links between resources in a hypermedia system.

As a further enhancement, we could aim for an adaptive hypermedia system which provides context-dependent link information based on individual user characteristics or user roles. For example, in the case of the Blue Planet course material, two roles of *student* and *tutor* could be used to determine the links available to the user. The tutor might get supplementary information with more detailed explanations or possibly with information about typical student difficulties. If the Blue Planet material were used, not only by university students, but also by schools, then the age of the student could also be used to determine the links.

Through the use of the Blue Planet example, we have presented some of the requirements for highly-connected mixed-media information systems. These include flexible linking schemes that enable users to navigate freely within the mixed-media information space and integration of application databases to enrich that space. Our approach was to develop a general link model and framework for mixed-media that could be extended in a natural way to support particular media types such as paper. In the next section, we introduce the general framework and after that describe its extension to support digitally augmented paper.

We conclude this section by noting that mixed-media integration and cross-media publishing is not just something of concern to the large commercial content publishers such as the BBC. It is something that most academics do in producing teaching materials such as lecture slides, handouts, manuals, tutorials, exercises, demonstrator applications and course websites. The vision is that instructors could provide *pre-authored links* between materials that they publish, while students could *dynamically author links* between the materials provided and also to any other materials that they find relevant.

3. iServer Framework

Many existing mixed-media integration systems are limited in integrating only two different types of media. We strongly believe that the key to a highly integrated solution lies in the introduction of an abstract layer for *generic link management*, which then can then be successively extended by new types of media and therefore leads to a true integration across multiple types of different resources.

Our aim therefore was to develop a general integration server architecture, called iServer, based on a general object-oriented link model as indicated in Fig. 3. To achieve maximum generality, the iServer link model provides the necessary functionality to link *entities*, where an entity can be an entire information resource such as an image file or an element within a resource such as a word within a printed document or a time sequence within a video file. An entity can equally be either an anchor or a target of one or more links and links can themselves be entities, which allows for links over links.

We therefore show in Fig. 3, a set of `Entities` which can be linked together and two subsets `Selectors` and `Resources` representing entities

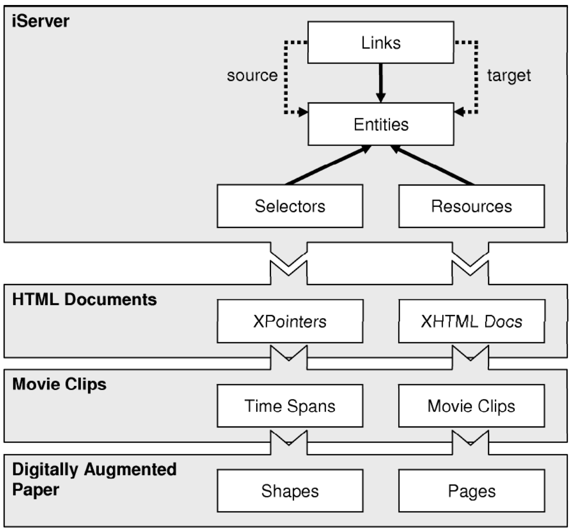


Figure 3. Extensible Link Server

which are elements within a resource and entities which are entire resources, respectively. For a particular media type, we can extend the framework by introducing a component that defines selectors and resources for that media type. In Fig. 3, we show a component for *movie clips* where a resource is a movie clip and a selector may be a time span or some spatial information. Further, we show a component for *HTML documents* where a resource is an XHTML document and a selector is an XPointer expression. This notion of selector is similar to that of reference objects in the open hypermedia model FOHM [21].

In the case of digitally augmented paper, the resources are pages and the selectors are shapes that define *active areas* within these pages. This is shown in Fig. 4 which highlights more details of the iServer link model in the upper part and the extension for digitally augmented paper in the lower part. We will discuss the upper part in the rest of this section and then discuss the lower part in the next section where we describe our integration of printed and digital media in detail.

We use the OM model [22] to describe the iServer architecture since the entire system is based on this model and its associated OMS data management frameworks [23]. The OM model is a generic, object-oriented model that supports information modelling through a two-level structure of classification and typing. It differs from commonly used object models such as UML in that it is, not only intended for system design, but also as an operational model for data management. The OM model therefore defines a full operational model over objects, collections and associations as well as constructs for their definition.

The shaded rectangles in Fig. 4 denote collections of objects and the shaded ovals represent associations over collections. A collection is a semantic grouping of objects (classification) and its name is specified as the central text. In the shaded part of the collection box, the member type is specified. This member type serves both as a constraint on membership of a collection and also as a default view of objects as accessed in the context of that collection. Whereas the collection deals with the classification of objects, the types deal with their implementation.

The core link concept of the iServer schema is denoted by the collection `Links`. The `HasSource` and `HasTarget` associations relate link objects to members of collection `Entities` which is partitioned into three subcollections `Links`, `Selectors` and `Resources`. `Links` can be associated with many `Entities` through both the `HasSource` and `HasTarget` associations and this means that our model is capable of supporting links with multiple source anchors as well as links with multiple targets (multi-headed links). The fact that `Links` is a subcollection of `Entities` enables links to be defined over links, thereby allowing links to be annotated with supplementary information.

As described previously, `Resources` and `Selectors` have to be implemented for specific media types. `Resources` represent the basic information components for a specific media whereas `Selectors` allow parts of `Resources` to be addressed and therefore control the granularity of link anchors and targets. The `RefersTo` association represents the fact that a selector is always associated with exactly one resource, whereas each resource could have more than one selector that references it.

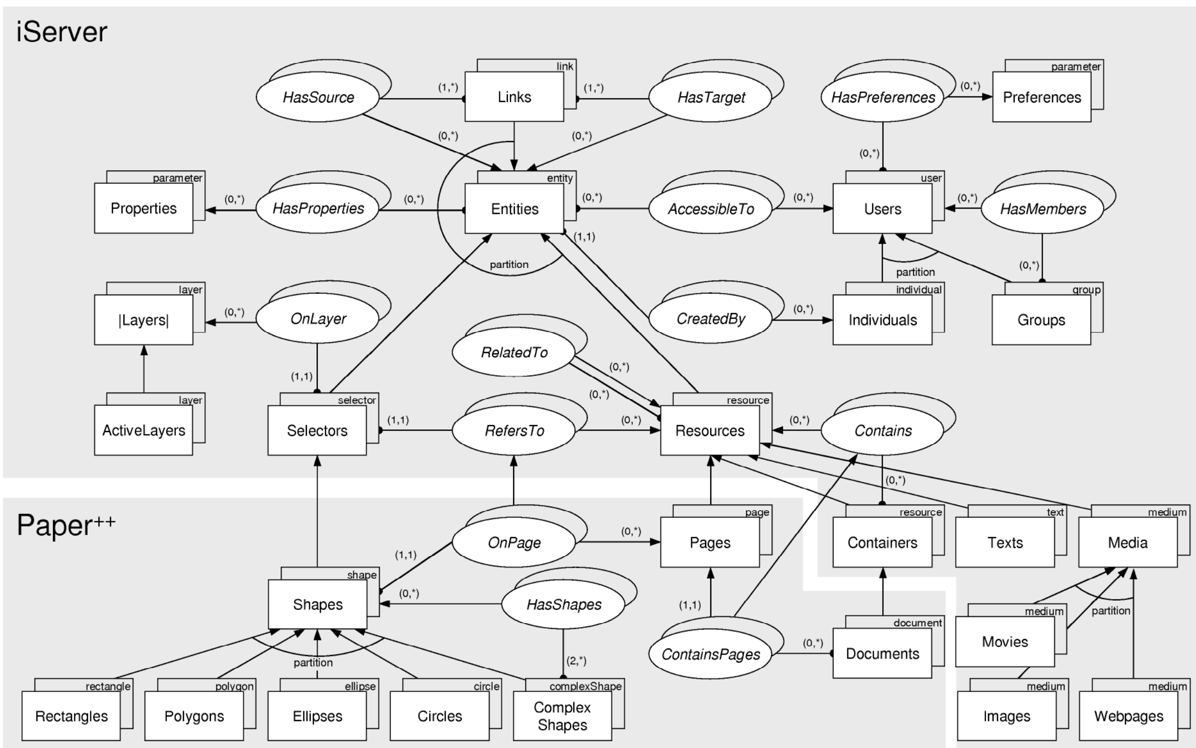


Figure 4. iServer with Paper^{++} extension

`Selectors` are associated with `Layers` to allow for a layering scheme that enables us to handle overlapping links and also context-dependent linking. We defer discussion of this to the next section where we motivate it and describe how it operates in terms of digitally augmented paper.

Each object in `Links`, `Resources` and `Selectors` is associated to exactly one creator and a set of users who can access it. By integrating a user model into the core link model, we are able to support the sharing of dynamically authored links as well as user-adaptive linking.

The lower right portion of the iServer schema deals with the classification and association of `Resources`. It is important to point out that a resource may be a *container* as well as an elementary resource such as a text, image or video. This enables us to deal with documents (printed or electronic) as complex resources which contain other resources. It also enables parts of images as well as whole images to be referenced. For example, in Fig. 5 of the next section, we show parts of an image linked to various multimedia files.

Our approach of developing a generic link management framework based on a link metamodel led naturally to a very general and extensible link model that supports concepts of advanced hypermedia systems. Most link models used in hypermedia systems tend to focus on only one or two of the advanced link concepts that we have introduced and few link services support nested links. For example, *ThirdVoice* as well as Hyper-G's *Harmony* browser [24] support overlapping links but they do not provide any functionality to explicitly define the semantics of such overlapping links. The XML Linking Language (XLink) [25] can be used to define external multiheaded links but does not support user management and multi-layered links.

Of course we are aware of all the results achieved by the open hypermedia research community on enabling access to link services by third party applications and the integration of different link services. The iServer framework is a new link service enabling the integration of physical and digital information. We see two possibilities how iServer can interface with other link services. The first one has been elaborated by the open hypermedia community in form of a lingua franca defined by the Open Hypermedia Protocol (OHP) and an Open Hypermedia Reference Architecture (OHRA) which should help to integrate existing link services [26]. Second, an existing link service can always been regarded as a new iServer resource type and be integrated by implementing a service specific plug-in (resource and selector) as shown in Fig. 3.

The generality that we achieved may in part be contributed to the use of the OM model. Its association construct enables relationships to be classified and manipulated directly which becomes particularly useful in supporting link management for hypermedia browsing. Further, since these associations are bi-directional, all associations used within the iServer framework are automatically bi-directional and it was therefore as easy to access all anchors for a given target object as to access the set of targets for a given anchor object. The separation of these references as associations also made it natural to separate the concept of a link from specific anchors and targets and allow for links with multiple anchors and/or multiple targets. For example, we may want to link a concept to its definition. But that concept may have many occurrences (and possibly representations). For instance, the concept of *coral* is represented many times in the Blue Planet text, either as an occurrence of the word 'coral' in the text or also as an image. We have a number of resources that serve to explain what coral is such as a textual definition or a video clip. A specific anchor and target represent just one relationship instance of what is a single *semantic link* between the concept and its definition. Note that if a user activates a multi-headed link, he will no longer get direct access to a target resource. Rather, a specific resource has to be chosen from the list of available link targets based on the resource's name.

To conclude this section, we mention that a mixed-media integration service should also be very flexible in supporting various forms of output channels for information visualisation. Again, the system should be open and extensible to support new kinds of output devices as soon as they come to the market. This is not a new issue in web publishing where content management systems are used to support various types of customised output channels. To achieve the same flexibility, the iServer framework uses concepts similar to those of content management systems to deploy the link information in the appropriate format [27].

4. Extension of iServer for Digitally Augmented Paper

Having presented the core link model of the iServer framework, we now turn to describe how we extended this very general link infrastructure to support digitally augmented paper. As already mentioned in Sect. 2, it would be good to have a more flexible means of addressing parts of paper than unique resource IDs provided by standard barcodes or some alternative form of encoding. Therefore, we wanted to shift to a solution that binds elements of printed material, such as pieces of text or images, to digital resources by means of invisible position encodings. This implies that entire pages are encoded with position information. For example, in the European project Paper^{++} [28], we used a specially designed grid of barcodes printed using invisible conductive ink. A number of other technical solutions have been proposed to enable position detection on paper [14,29] and these could be equally well used within our framework.

Based on some form of invisible position encoding, we can associate our content with logically defined *active areas* of the page as indicated in Fig. 5. An active area can be defined as one of a number of simple geometrical shapes or as a complex shape composed from multiple other shapes. A shape becomes a tool to address parts of a single page and is the selector used for printed page resources. Therefore, in the case of digitally augmented paper, the type `page` becomes a specific implementation (subclass) of the more general `resource` type whereas the `shape` is a media-specific implementation of a `selector` type (see lower part of Fig. 3). The inductive reader device will deliver an (x, y) page coordinate and the iServer will then determine which, if any, active area contains this point and then return the resource which is bound to the selected active area. The default Paper^{++} setup includes one or more digitally augmented paper documents, an inductive reader for position detection and a laptop computer to display the results.

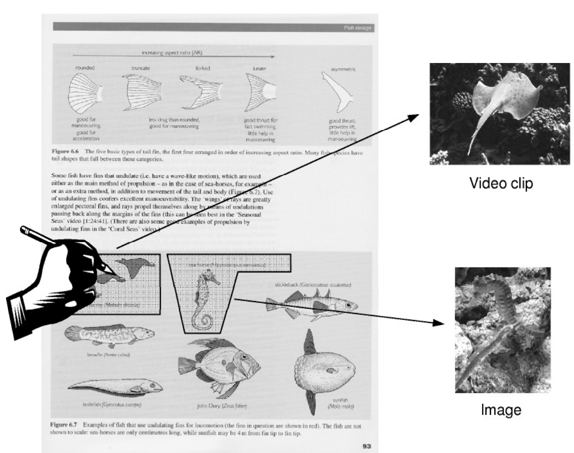


Figure 5. Active areas

The shape concept allows us to address active areas of a page, but it is not yet possible to flexibly control the granularity of the link anchors on paper. For instance, we may be interested in detailed information about a certain part of the body of a fish such as the fin, or a word concept printed within an image. It is also possible that areas may overlap as arises when a printed text flows over an image. This requires a further extension to the information model to support a layering scheme.

Each selector, and hence each shape, is associated with a specific logical layer and there can be no overlapping shapes within a single layer. As a consequence of this, overlapping shapes must belong to different page layers and, in the case that the selected position lies within two or more shapes, the shape within the uppermost layer is selected. In addition, specific layers may be activated and deactivated which enables us to generate context dependent results by binding a particular position on a page to different resources according to the current set of active layers. For example, this could be used to provide a "zoom in" effect on images if we were to define the shapes and layers such that repeated selection of a position on a page caused the uppermost layers to be deactivated in turn, thereby moving down to smaller image parts defined on the lower levels. The zoom in/out functionality entails a whole range of new interaction and user interface possibilities in working with static paper, where the digital media can be used to give feedback about the context (current layer) within an image.

To model this multi-layer functionality, we represent the layers as a totally ordered collection `|Layers|` (indicated by the two vertical bars) with a subcollection `ActiveLayers`. Note that there is support for collections with different behaviours (sets, bags, rankings and sequences) in OM and its associated systems. The association `OnLayer` relates each member of `Shapes` to exactly one member of collection `|Layers|`.

As stated previously, the OM model has a full algebra over objects, collections and associations. Without going into detail of the OM query language AQL, we show below a query that, given a point (x, y), will return the uppermost shape of the active layer set that contains that point.

```
first(domain((OnLayer rr ActiveLayers)
  dr (all S in Shapes having S.contains(x,y))
))
```

First, two selections are applied to the association `OnLayer`: we select pairs where the second element belongs to `ActiveLayers` and, in a second step, we further select those pairs where the first element is a member of `Shapes` where the method `contains` returns `true` for the input parameters x and y. Having carried out this reduction of the `OnLayer` association, we then take the `domain` which gives the collection of shapes containing the point (x, y) and belonging to an active layer. Since `OnLayer` is a total ordering that respects the ordering of `Layers`, the `domain` operation will maintain that ordering and `first` will select the uppermost shape as required. Details of the algebra and query language are given in [30] whereas [31] provides additional information about multi-layered linking.

So far we have achieved a mapping of elements of a printed document to digital resources, but we still do not have the capability to perform an inverse mapping from the digital world to the printed world. The problem is that we have insufficient information about the context of an embedded link within a document. It is not enough to know the coordinates of a shape within a page that define an active area. We must also know the page and the document that contains that area. We therefore introduce information about `Documents` and `Pages` as shown in Fig. 3. Note that both `Pages` and `Documents` are subcollections of `Resources` and the association `ContainsPages` between `Documents` and `Pages` is a specialisation (subassociation) of the `Contains` association between `Containers` and `Resources`. This means that it is possible to deal with entire printed documents as a resource as well as individual pages.

While this is simple to model, we must think about what this means in terms of the underlying technologies. The encoding system must be amended to include information about the document and page number. With current encoding schemes, it would be difficult to include globally unique document IDs within a grid-based page position encoding scheme. Within the Paper^{++} project, we have encoded page and coordinate information within the page encoding scheme and handle document IDs separately through either the use of RFID tags or special printed document encodings on the cover page.

With the additional contextual information of document and page, we are able to map digital resources to active areas within documents. For example, we could easily find all the places where a given video or term (printed word or phrase) is used within a document set. This now gives us a basic mapping back and forth between printed and digital media. For example, in Fig. 6, we show an application that we developed for the Paper^{++} project where research publications in

Figure 6. Links from digital content to paper

printed form could be linked to digital information about various research projects and technologies. If a technology is referenced in a paper, this enables us to link to digital information about that technology and projects that use it. In addition, we can find places in other papers where that technology is referenced. The screenshot in Fig. 6 shows the result of requesting all link anchors for the digital concept representing an iPAQ pocket PC. As a response, the system tells us that the iPAQ concept has been linked from two different physical documents, shows us the title of the document and the corresponding page on which the link anchor has been defined and, further, highlights the active region on a digital representation of that page. These links back to physical content are “manual” and the user has to actively fetch the corresponding document and track the link target on the appropriate page.

Note that these back references to printed documents are not just another piece of text within the database giving, for example, a document title and a page number. In fact, the links back to physical paper are built dynamically based on meta information about other documents bound to the same information object. This meta information is already present in the information

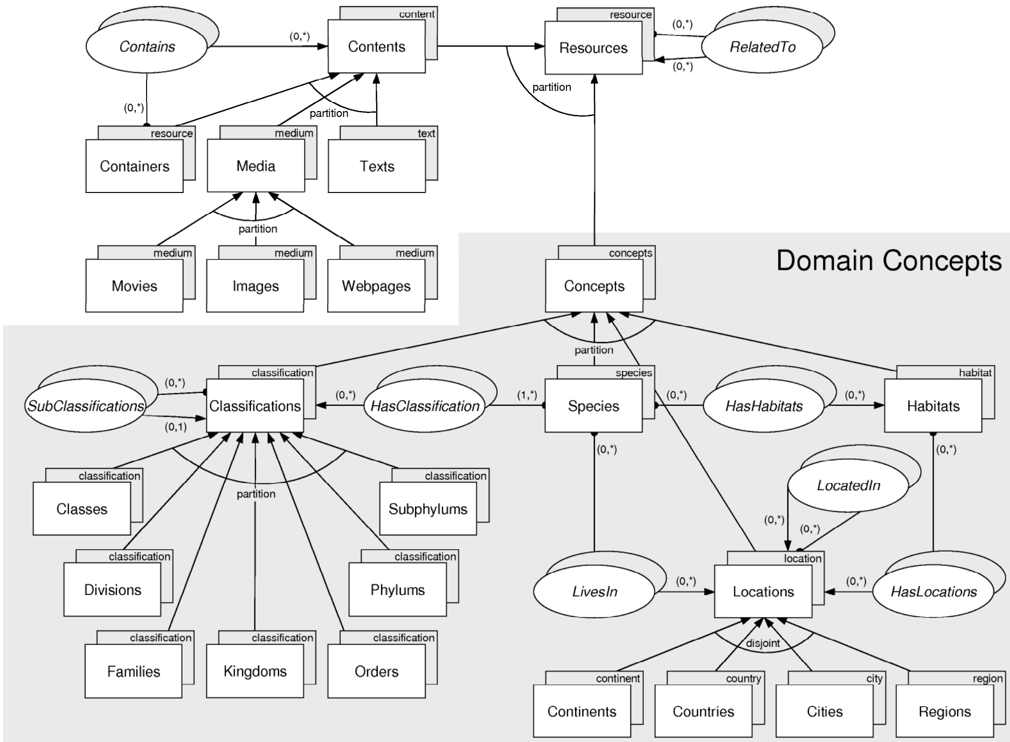


Figure 7. Semantically rich application domain concepts

model (bidirectional associations) and therefore no additional authoring is required.

5. Integrating Application Databases

At this point we have achieved a relatively simple two-way linking of printed and digital resources. But can we really say that we have achieved an integration of printed and digital information? The answer is "no" since there is no information about the semantic content of these resources. On the printed side, our objects of interest are simply areas of a page. On the digital side, we simply have content such as texts, images or movies without these being related to any conceptual information about a specific application domain.

To provide a richer information environment, we could introduce information concepts into the server by extending the schema to cover concepts about the application domain as shown in Fig. 7.

We have now partitioned `Resources` into `Contents` to represent the basic text and media files as before and a new collection `Concepts` to cover the representation of application specific information. Essentially, `Concepts` is a way of integrating an application database into the information server of the cross-media publishing framework. The application domain of the schema presented in Fig. 7 is a nature database with information about species and their habitats.

With the introduction of an application schema which effectively links different concepts of the application domain, note that we can also introduce more powerful forms of derived linking of printed materials. Assume that there is a link from an element of one document to a particular species object. Further assume a link from a second document such as an atlas to a location object. Then through the associations `HasHabitats` and `HasLocations`, a user interested in that species and its habitat can be directed to one or more places within the atlas. Derived linking may be particularly powerful in the case of open authoring systems where users may dynamically add their own links from printed documents to digital resources and links can be shared within user communities.

This leads us to the final aspects of the iServer framework that we want to consider here, namely personalisation and customisation. We have already alluded to the use of layering as a potential means of presenting context-dependent information through the dynamic activation and deactivation of layers. In addition, we may want to present different materials to different users according to preference or role as discussed for the Blue Planet material in Sect. 2. For example, in the case of a printed encyclopaedia, we may want to present different links and content dependent on the age of the user. Typically, children prefer less detailed textual information and more images and audio. For educational scenarios, we may want to present different information according to whether the user is an instructor or a student. In particular, links from an exercise sheet may link to hints in the case of a student and detailed solutions in the case of an instructor.

For this reason, it is important to model users within the system and for that we introduce into the iServer a special user concept as described in Sect. 3. However, note that the user information may be shared across applications and hence we show it as a separate component in Fig. 8.

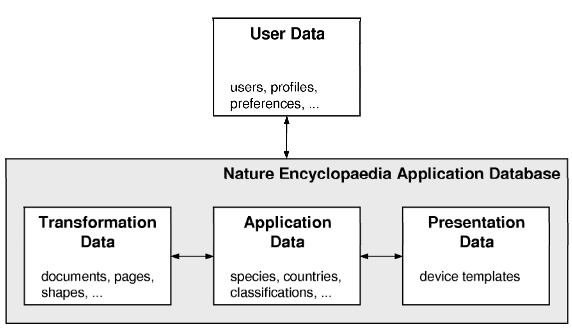


Figure 8. Information server

Here we show the main components of the information server framework. Generally, for each application, there is a *Transformation Component* that defines active areas of the document set and their binding to resources, an *Application Component* that is the domain-specific database and also a *Presentation Component* which manages templates to enable client-dependent delivery of content. In Fig. 8, we show the components developed for a first demonstrator application that linked together printed and digital information from a children's nature encyclopaedia. This application was accessible from a range of devices including standard desktop browsers and PDAs as described in [11]. The *User Component* deals with user profiles and preferences may be shared by a number of applications, although some parts of it may be application-specific.

As outlined in the discussion of the Blue Planet materials in Sect. 2, the introduction of an application database can potentially provide readers of printed material with access to a wealth of related digital materials. However, like all database development projects, it needs significant investment to do it well. Developing a full application database requires domain expertise and lots of data input. Therefore this may not be the solution for all forms of cross-media publishing. However, it is important to point out the potential long-term rewards of such an investment, especially when one considers reusability. Even within the Paper^{++} project, we found that we could reuse all or part of application models. As mentioned previously, our first demonstrator application was based on a nature encyclopaedia and this enabled us to carry out preliminary user studies [32]. When we came to examine the BBC Blue Planet material, we found, of course, that the core application schema remained much the same and the difference lay in the data.

Since it would be possible for the *Application Component* to be shared by applications, it might be worth the long-term investment for major publishers with interests in mixed-media publishing to develop databases that would provide a source of material for a number of applications. For example, the BBC are well-known for the high standard of their nature documentary series and over the years there have been many. A well-defined and extensive nature database could be used for all of these programmes and, of course, would provide a rich source of material that the BBC itself could use in the planning and development of their programmes. In fact, according to our sources, they are in the process of developing such a database and this could provide a valuable source of material if integrated into their mixed-media publishing activities in the future.

In other application domains, it may be the case that databases are already in existence that could support their mixed-media publishing. It is therefore also important to note that the application component could be an external database source integrated into the framework, rather than a specially designed component. Various commonly used database integration techniques could be used such as representing concept resources in the iServer framework as objects with methods that request the appropriate data from the external database.

To conclude this section, we point out that the application database could be any form of database and might even be a domain ontology. The framework could therefore also be used to support metadata annotation as proposed for the semantic web [33] and specifically its use to generate links between documents in open hypermedia systems [34,35]. In fact, in some cases we may want to support links to both application data and metadata. Since all information (data, metadata and system data) is stored in OMS as objects, it is in fact straightforward to provide all of these possibilities within the iServer framework.

6. Authoring and Publishing Tools

When considering the activity of authoring mixed-media material, there are a number of fundamental factors to take into account. These factors deal with issues such as the intended usage of the system and the source of materials. They will greatly influence the types of authoring to be supported and hence the approaches and tools to be used. While it is beyond the scope of this paper to describe any particular authoring tool or publishing framework in detail, we want at least to briefly indicate some of the approaches that we are investigating.

The first factor to consider is whether the content is already available in the form of digital and printed materials. If so, the main authoring activity is *link authoring* and a tool is required to support the creation of links between the existing content elements. The second factor is whether only the publishers can author links or also the users of the material in which case *dynamic link authoring* must be supported. If users can author their own links, the next question is whether there is an *open link authoring* scheme, in which users can link arbitrary materials and not only those provided by a single publisher. For example, a student may wish to link concepts in a set of lecture notes to text books or to websites. Clearly, the ability to freely create links between arbitrary printed materials implies a major shift to a future world of interactive paper.

Developing technologies that enable invisible, or nearly invisible, encodings brings us back to the issue of how users know where links are located within a page. In terms of published content with embedded active links, highlighting can be used or even recognised conventions that have arisen from the use of web browsers where users have come to expect content items such as headers, images and underlined or blue text to be possible links. We have also experimented with the use of audio feedback to inform users that they have activated links. With the further development of reader devices, other forms of audio or force feedback from the reader device itself could be used to inform a user of the existence of a link on rollover. Another possibility would be to combine a highlighter pen with the reader device as described in [8].

Within the Paper^{++} project, we investigated existing authoring "solutions" for digitally augmented paper and these were found to be restricted in flexibility. For example, typically they handled only rectangular shapes and lacked support for context-sensitive information such as user-dependent anchors or personalised links [36]. It was therefore necessary to develop our own authoring tool that could support the functionality offered by the iServer framework. Our tool supports any kind of shapes and multi-layer authoring for complex figures based on an optional digital representation of the document. Figure 9 shows the authoring tool with its panels for layer management, user management and the definition of active areas on a document page. The tool also enables the authoring of multi-user documents which contain anchors (shapes) linking to different resources based on the user currently working with the document. Further, we are investigating a better integration of the reading device into the overall authoring process. The user should not have to define the active page areas (anchors) by marking them on a digital on-screen representation of the document, but rather be able to generate them directly on the physical paper by using the reader to draw a shape's outline.

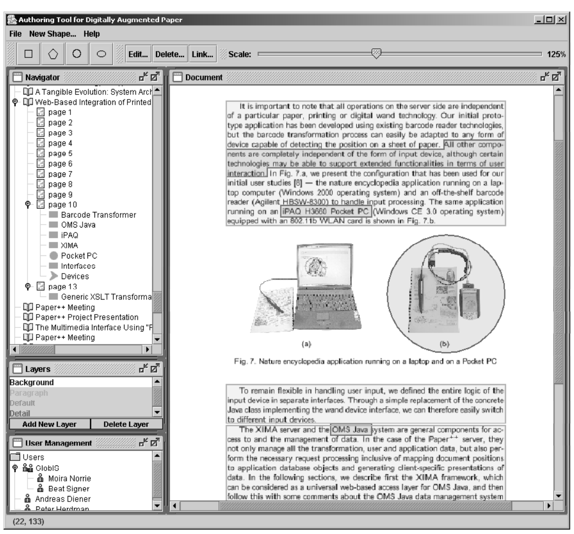


Figure 9. Paper^{++} authoring tool

The explicit authoring of links can certainly be time-consuming and it may be based on the knowledge or experience of a single individual. Especially in the case of open, dynamic link authoring systems, it is desirable to share links among communities of users, thereby enabling users to benefit from the knowledge of others. In addition, user profiling and access paths can be used to generate virtual trails, suggesting links to users based either on their own or the community's experiences similar to the ideas described in [37]. These are topics that we have partly addressed in the context of other projects (e.g. a personal assistant for access to web databases [38]) and we are currently integrating these ideas into the Paper^{++} information server.

So far, we have considered only link authoring. If the content does not already exist, then the authoring activity will consist of both *content authoring* and *link authoring*. In this case, the content can be developed with the resulting hypermedia system in mind and tools can be used that generate both printed and digital documents along with the links between them. In such a case, instead of using traditional authoring tools such as a word processor, a content management approach of the form now prevalent in website engineering could be employed. For example, within our research group, we have integrated concepts for web publishing into OMS systems that enable us to combine features of content management with that of web databases. This means that we are not only capable of managing the content, structure and presentation of web documents, but also deriving that content from semantic entities of an application database thereby ensuring the semantic consistency of web sites [39]. A major requirement of such systems is multi-channel delivery and we are now able to exploit these systems to generate, not only digital documents such as HTML for web browsers, but also printed documents with embedded active links.

We have briefly outlined the various forms of authoring activity that need to be supported and indicated the sorts of tools and technologies that are required. However, it is clear that, in many cases, the different forms may all be required within a single application domain. For example, in a university learning environment, the instructors may want to generate on-line and printed material which is linked together, but additionally students will want to add and share links both within this set of materials and also to external sources.

7. Concluding Remarks

The field of cross-media publishing for digitally augmented paper is rapidly evolving as the enabling technologies begin to emerge into the marketplace. To date, a study of the literature reveals that the "information" aspect often tends to be ignored or underplayed in favour of the media technologies. Proposed solutions tend to be based on primitive information models that in many cases perform a simple, one-way mapping from paper to digital media.

In contrast, we have focussed on the information management issues and have developed a general framework for mixed-media information systems based on a generic, extensible link model to support all types of media. Our system for digitally augmented paper was then based on a specific extension of the framework that enables active areas of paper to be linked to digital resources and vice versa. Thus, the framework enables paper to be introduced as a first-class medium into open hypermedia systems.

Such a framework opens up many new possibilities for forms of information interaction and also radical new approaches to information publishing. Its realisation should therefore not be seen as defining the end of a research project, but rather as defining the beginning of a new research area.

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. K. M. Anderson, R. N. Taylor, and Jr. E. J. Whitehead. Chimera: Hypermedia for Heterogeneous Software Development Environments. *ACM Transactions on Information Systems*, 18(3):211–245, 2000.
2. W. Hall, H. C. Davis, and G. Hutchings. *Rethinking Hypermedia: The Microcosm Approach*. Kluwer Academic Publishers, May 1996.
3. H. Maurer. *Hyperwave: The Next Generation Web Solution*. Addison Wesley, 1996.
4. T. Kindberg, J. Barton, J. Morgan, G. Becker, D. Caswell, P. Debaty, G. Gopal, M. Frid, V. Krishnan, H. Morris, J. Schettino, B. Serra, and M. Spasojevic. People, Places, Things: Web Presence for the Real World. In *Proceedings of WMCSA'2000, 3rd IEEE Workshop on Mobile Computing Systems and Applications*, Monterey, USA, December 2000.
5. H. Ishii and B. Ullmer. Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms. In *Proceedings of ACM CHI'97, Conference on Human Factors in Computing Systems*, volume 1, pages 234–241, March 1997.
6. 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, pages 370–377, May 1999.
7. A. J. Sellen and R. Harper. *The Myth of the Paperless Office*. MIT Press, November 2001.
8. 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, pages 327–334, March 1997.
9. M. Dymetman and M. Copperman. Intelligent Paper. In *Proceedings of EP'98, 7th International Conference on Electronic Publishing, Document Manipulation, and Typography*, pages 392–406, April 1998.
10. D. L. Hecht. Printed Embedded Data Graphical User Interfaces. *IEEE Computer*, 34(3):47–55, March 2001.
11. M. C. Norrie and B. Signer. Web-Based Integration of Printed and Digital Information. *Lecture Notes in Computer Science*, 2590:200–219, 2003.
12. The Blue Planet Website, BBC, http://www.bbc.co.uk/nature/blueplanet/.
13. The Open University, http://www.open.ac.uk/.
14. Anoto AB, http://www.anoto.com.

15. MyPen, C-CHANNEL AG, http://www.c-channel.ch.
16. Logitech io Personal Digital Pen, Logitech Inc., http://www.logitech.com.
17. Nokia Digital Pen, Nokia, http://www.nokia.com.
18. PaperLink, code-it AG, http://www.code-it.ch.
19. P. Ljungstrand, J. Redström, and L. E. Holmquist. WebStickers: Using Physical Tokens to Access, Manage and Share Bookmarks to the Web. In *Proceedings of DARE'2000, Designing Augmented Reality Environments*, Elsinore, Denmark, April 2000.
20. NeoMedia Technologies. IDOCs Linking the Worlds of Print and Electronic Media. Technical report, NeoMedia Technologies, September 1998.
21. D. Millard, L. Moreau, H. Davis, and S. Reich. FOHM: A Fundamental Open Hypertext Model for Investigating Interoperability between Hypertext Domains. In *Proceedings of ACM Hypertext 2000, Conference on Hypertext and Hypermedia*, May 2000.
22. M. C. Norrie. An Extended Entity-Relationship Approach to Data Management in Object-Oriented Systems. In *Proceedings of ER'93, 12th International Conference on the Entity-Relationship Approach*, Arlington, USA, December 1993.
23. A. Kobler and M. C. Norrie. OMS Java: Lessons Learned from Building a Multi-Tier Object Management Framework. In *Proceedings of OOPSLA'99, Workshop on Java and Databases: Persistence Options*, Denver, USA, January 1999.
24. K. Andrews. *Browsing, Building, and Beholding Cyberspace: New Approaches to the Navigation, Construction, and Visualisation of Hypermedia on the Internet*. PhD thesis, Graz University of Technology, 1996.
25. S. J. DeRose. XML Linking. *ACM Computing Surveys*, 31(4), December 1999.
26. S. Goose, A. Lewis, and H. Davis. OHRA: Towards an Open Hypermedia Reference Architecture and a Migration Path for Existing Systems. *Journal of Digital Information*, 1(2), December 1997.
27. M. C. Norrie and B. Signer. Web-Based Integration of Printed and Digital Information. In *Proceedings of DIWeb'02, 2nd International Workshop on Data Integration over the Web*, Toronto, Canada, May 2002.
28. Paper^{++} Project, IST-2000-26130, Disappearing Computer Initiative, http://www.paperplusplus.net.
29. C. Kafka. DataGlyphs Bridge Paper and Digital Worlds. *Docu World*, 1998.
30. A. Würgler. *OMS Development Framework : Rapid Prototyping for Object-Oriented Databases*. PhD thesis, ETH Zurich, 2000.
31. B. Signer and M. C. Norrie. Multi-Layered Cross-Media Linking. In *Proceedings of ACM Hypertext 2003, 14th Conference on Hypertext and Hypermedia*, Nottingham, UK, August 2003.
32. D. Frohlich, E. Tallyn, N. Linketscher, B. Signer, and G. Adams. Reading Augmented Paper: Children's Experiences from a Simulation Study. Technical Report HPL-2001-308, HP Labs, 2001.
33. T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. *Scientific American*, May 2001.
34. S. Bechhofer and C. Goble. Towards Annotation using DAML+OIL. In *Proceedings of Workshop on Knowledge Markup and Semantic Annotation*, Victoria B.C., Canada, October 2001.
35. J. Kahan, M.-R. Koivunen, E. Prud'Hommeaux, and R. R. Swick. Annotea: An Open RDF Infrastructure for Shared Web Annotations. In *Proceedings of WWW10, 10th International World Wide Web Conference*, Hong Kong, May 2001.
36. I. Siio, T. Masui, and K. Fukuchi. Real-world Interaction using the FieldMouse. In *Proceedings of UIST'99, 12th Annual ACM Symposium on User Interface Software and Technology*, November 1999.
37. V. Bush. As We May Think. *Atlantic Monthly*, July 1945.
38. B. Signer, A. Erni, and M. C. Norrie. A Personal Assistant for Web Database Caching. In *Proceedings of CAiSE'2000, 12th Inter-*

national Conference on Advanced Information Systems Engineering, Stockholm, Sweden, June 2000.

39. M. C. Norrie and A. Palinginis. Versions for Context Dependent Information Services. In *Proceedings of COOPIS 2003, 11th International Conference on Cooperative Information Systems*, Catania, Italy, November 2003.