A Model and Architecture for Open Cross-Media Annotation and Link Services

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

[a]*Vrije Universiteit Brussel, Pleinlaan 2, 1050 Brussels, Belgium*
[b]*ETH Zurich, 8092 Zurich, Switzerland*

Abstract

Over the last decade, we have seen a tremendous increase in the number of digital media types that we deal with as part of our daily work. While the Web with its linking functionality was originally designed for organising information in the form of HTML documents containing embedded media such as images, movies and sounds, the underlying hypertext model is not flexible enough to deal with new media types. A flexible link and annotation service should not only support a growing set of digital media types, but also pay attention to emerging possibilities for linking and integrating the physical environment with digital information spaces in the form of augmented reality environments. The successful implementation of these so-called cross-media information spaces, where different types of digital information get linked and integrated with physical entities, demands for a rethinking of models and architectures for extensible and scalable cross-media annotation and linking. In this paper, we present our general model for open cross-media annotation and link services and highlight how this model enabled the realisation of an extensible cross-media architecture. We further introduce the concept of open cross-media information spaces where the integration of new media types on the data level as well as on the visualisation level is supported via a resource plug-in mechanism.

Keywords: cross-media information space, open cross-media service, annotation, linking, digital library

1. Introduction

With the rapid growth of Web 2.0 communities, many users are no longer simply passive readers of information published on the Web but have become actively involved in the information management process by creating new content or annotating existing resources. While web technologies have enabled the large-scale and low-cost sharing of information, link and annotation services allow users to integrate and augment that information in an ad-hoc manner without explicit pre-defined integration schemas or the need to have a local copy of that information. If the link and annotation metadata is stored separately from the original resources, the author does not need any write access to the original content. This allows communities of users to build an overlay knowledge layer on top of existing web resources through various forms of link and annotation services. As a result, the idea of external link metadata, as introduced by the hypertext community in the form of dedicated link servers, has finally found its manifestation in a number of widely used Web 2.0 applications.

The opening of information resources to third-party contributors has also been recognised by the digital library community as a way of enriching existing content with community-based annotations and associations to supplementary external resources. By bridging the gap between content managed within a digital library system and digital information available outside of the library, as well as enabling links and annotations across digital library systems, external annotation and link services may contribute to the integration of content managed by different digital libraries.

The potential of knowledge sharing through collaborative links and annotations can only be fully exploited if a general and sustainable *link and annotation fabric* can be established to ensure that links and annotations persist over time and can be reused and extended by future applications. Therefore, some common standards and guidelines are required to make different link and annotation services interoperable rather than producing isolated and proprietary solutions. In the context of the Web, we have already seen first efforts to establish specific annotation standards, including the one defined by the Annotea[1] framework. The XML Linking Language (XLink) [1] is an attempt to introduce a more powerful resource linking standard that goes beyond the simple unidirectional and embedded link concept offered by HTML. The digital library community has also tried to establish annotation standards by defining digital library reference models which include annotations as information objects.

However, in addition to specifying common link models and standards, it is necessary to define a flexible and extensible reference architecture capable of supporting any form of cross-media links and annotations. It is no longer sufficient to support only textual or multimedia annotations to a fixed set of resource types. The Web is a platform with a rich and continuously evolving set of multimedia types. Therefore, it is important to ensure that link and annotation services can be extended to cater for new media types at the data level as well as by integrating them into cross-media link authoring and browser tools. While we can see some emerging solutions for extensible annotation data models, currently there exist no solutions to achieve

[1] http://www.w3.org/2001/Annotea/

The final publication is available at http://dx.doi.org/10.1016/j.is.2010.08.002

the same flexibility and extensibility on the application and visualisations level. This implies that any extension of the underlying annotation model to support a new media type requires major changes and extensions to the corresponding annotation authoring tool and browser. The problem gets even worse if there exist multiple applications and annotation browsers that rely on a common annotation model since all these tools have to be modified individually to support any new media type. In this paper, we present an extensible cross-media link and annotation architecture to address these problems in terms of extensibility on the application and visualisation level along with the general cross-media link model it is based on.

We start in Section 2 by providing an overview of existing annotation and link services. In Section 3 we introduce the concept of open cross-media annotation solutions and discuss some of their requirements in terms of extensibility on both the model and architecture level. We then introduce our cross-media annotation model in Section 4, discussing how it supports extensibility and comparing its main features with existing annotation proposals. Some more details of how to realise an annotation service based on the proposed model and architecture are provided in Section 5. An evaluation of our cross-media annotation and link solution is provided in Section 6 and concluding remarks are given in Section 7.

2. Existing annotation and link services

Before discussing different solutions for content annotation and linking, we address the question of what the difference is between an annotation and a link or association with supplemental information. In our definition, the annotation process only "differs" from regular linking as implemented by various hypermedia systems through the fact that the creation of a new annotation often includes the content authoring of the annotation object itself. In contrast, link authoring usually creates associations between existing resources. We can therefore see annotation services as a specialised application of more general link services. This implies that we do not treat annotations as metadata but deal with them on the same level as any other information object. In this section, we therefore cover specific annotation services as well as more general hypermedia solutions. An annotation model introduces specific classes of information units, their relationships and how they can be used to build an annotation service. In the rest of the paper, we often mention just one of the two terms explicitly, assuming that the reader is aware of the similarities between the concept of a link and an annotation.

The Annotea [2] project developed by the World Wide Web Consortium (W3C) provides a framework for collaborative semantic bookmarking and the annotation of webpages via user defined tags (topics). Annotea makes use of the Extensible Markup Language (XML) in combination with the Resource Description Framework (RDF)[2] to manage annotation metadata about XML documents on separate annotation servers. The W3C's Amaya[3] browser and editor uses Annotea and the external metadata approach to annotate arbitrary webpages without modifying the original resources. The Amaya editor enables parts of an HTML document to be addressed based on XPointer[4] expressions which can then be annotated with arbitrary textual information. In addition to the Amaya browser, there exist plug-ins for different web browsers such as the Annozilla[5] extension for the Firefox web browser. While Annotea is limited in terms of the resources that can be annotated, the Co-Annotea system [3] provides an Annotea extension for annotating relationships between different mixed-media objects. Whereas it is easy to extend Annotea on the model level, the introduction of new media types requires major changes to the browser and authoring component since there is no common underlying extensible framework for the rendering of these mixed-media information spaces and the interaction with different media types is hard-coded in the visualisation tool.

While Annotea makes explicit assumptions about the type of documents to be annotated since parts of the document have to be addressable by the XPointer language, the Flexible Annotation Service Tool (FAST) [4] is extensible by providing a core annotation service with different gateways for specific information management systems. The gateway approach is a good mechanism to integrate the annotation service with different information management systems. However, FAST also does not explicitly deal with extensibility issues in terms of different media types on the annotation browser and application level. As we show in the next section, it is essential for an extensible cross-media annotation service that new media types can be introduced without major changes to existing applications.

An annotation service addressing parts of documents managed by a digital library system through the concept of *marks* is presented by Archer et al. [5]. Annotations can be stored either together with the document or in an external repository. Whereas this solution provides a flexible means of addressing specific document parts, it currently only supports textual annotations and it is unclear how easily other media types could be integrated in the future.

A fixed set of multimedia annotation types is supported by the web-based MADCOW [6] multimedia digital annotation system which uses a client-server architecture in combination with a browser plug-in. A good overview of MADCOW and other annotation solutions is given in [7]. While these systems can be extended on the model level to support new types of media, we will show that there is a lack of easy extensibility on the browser and editor level. In an optimal case, there should be a clear separation of concerns not only between the media-specific annotation details on the model level but also between a general visual annotation and link authoring tool and its components dealing with various types of annotation resources. As a contribution of this paper, we discuss some limitations of existing annotation tools. We then show that the same flexibility already offered by some annotation services on the model level,

[2] http://www.w3.org/RDF/

[3] http://www.w3.org/Amaya/

[4] http://www.w3.org/TR/xptr/

[5] http://annozilla.mozdev.org

has to be achieved on the information authoring and rendering level. We highlight how this kind of extensible cross-media annotation and link solution can be realised based on the presented architecture.

While the above systems enable annotations across different types of media, the interoperability of link services has been addressed by the open hypermedia community and different proposals including the Open Hypermedia Reference Architecture (OHRA) [8] have been made. The same comments given for the FAST annotation solution in terms of its limited extensibility on the information rendering level are also valid for OHRA and other open hypermedia architectures. The exchange of navigational link information between different open hypermedia systems is supported by the Open Hypermedia Protocol (OHP). OHP was defined using a Document Type Definition (DTD) document, resulting in a lack of details due to the limited expressiveness of the chosen "specification language". It is also questionable whether the Open Hypermedia Protocol can support the rich functionality offered by different open hypermedia systems. The Open Hypermedia Protocol has been extended by the Fundamental Open Hypertext Model (FOHM) [9] to offer a common data model and operations for navigational, taxonomic and spatial hypermedia. While FOHM offers a general data model for these three hypermedia domains, it does not cover any other existing hypermedia domains. The issue of limited extensibility due to a lack of structural abstractions necessary to support different hypermedia domains was further investigated by Component-based Open Hypermedia Systems (CB-OHS) [10]. A concept that is missing in many open hypermedia solutions is the idea of data and link ownership. A majority of open hypermedia systems do not deal with user management issues as part of the core hypertext model, but rather implement it on the application layer. It is evident that this complicates the controlled sharing of information, such as link metadata, across different applications of the same hypertext model, since each application might treat user-specific information in a slightly different way. The remark about the user management not forming part of a majority of hypertext models is also valid in the context of many annotation models.

Similar to some of the annotation services introduced earlier, open hypermedia systems store link metadata separate from the original resources on different link servers. The clear separation of link metadata and the corresponding resources alleviates the realisation of advanced link features such as bidirectional or overlapping links—concepts that were already introduced in Ted Nelson's Xanadu[6] project more than forty years ago. Furthermore, the use of external link servers in open hypermedia systems dissolves the clear distinction between the author of a link on the one hand and the user of the link metadata on the other hand. Since the authoring of link information no longer requires write access to the original resources, any user can create new link metadata and share it with other community members. Note that a link service can not only be provided as a third-party component, but can also form part of an entire software development suite tightly integrated with specific applications, as proposed by Sun's link service [11].

Over the last two decades, the hypertext community has realised a rich set of hypermedia systems. Some of the most widely known link servers include Microcosm [12], Chimera [13] and Hyperwave [14]. The lack of the possibility to manage links decoupled from the corresponding HTML documents in the original web infrastructure, led to the effort of developing a new XML Linking Language (XLink) [1, 15] to address some of these shortcomings. However, there is a weak acceptance and support for the XLink standard and only a few web browsers implement parts of the standard. For those browsers that offer XLink support, they only implement the basic XLink functionality which is just an alternative representation for embedded HTML links, missing any rich link functionality. The weak acceptance of the XLink standard in web engineering shows that it is not sufficient to define a powerful link model and that the model has to go hand in hand with an architecture and interfaces for the necessary authoring and rendering of information. One possible reason for the lack of XLink support in current web browsers might be the missing specification of requirements for an XLink browser and editing component.

While the original idea of the Web was to link digital resources in the form of HTML pages, emerging technologies such as RFID tags enable the tracking and unique addressing of physical resources. The vision of the *Internet of Things* foresees an integration of everyday objects with digital information and services accessible over the Internet. To achieve this goal, we need models and architectures that support the linking of digital and physical resources in a similar way as the presented link servers. Therefore, more recently, physical hypermedia models for bridging the physical and digital worlds have been proposed. For example, HyperReal [16] is a mixed reality model that introduces the concept of map components for managing geographical data. The question that arises is whether we not just inventing yet another hypermedia domain and whether these physical hypermedia models can be effectively integrated with existing link servers as well as information and services on the Web.

As illustrated in this section, there exist a wide variety of models and services for annotating and linking resources. While there have been some attempts to come up with reference models, for example FOHM, the majority of annotation and link models are proprietary solutions for specific domains, for exaple navigational hypertext, or even for individual applications. Some of the hypermedia models that have claimed to be general and extensible, have disappeared over time and been replaced by yet another hypermedia model. What makes this development slightly more critical is the fact that there is often little or no support for the migration of data from one hypermedia model to its "successor", resulting in a loss of data when switching to a new hypertext model. We believe that one factor causing this undesirable situation is a lack of well defined conceptual cross-media annotation and link models on which the architectures and implementations are based. This lack was also identified by the open hypermedia community when proposing

[6] http://xanadu.com

a globally distributed and collaborative model with a clear evolution path [8]. Existing models are often presented as a mix of conceptual, technical and architectural features without a clear separation of concerns. As a result, we see an obfuscation on the conceptual layer and often unnecessary limitations are introduced on the model layer due to some technical details of the intended implementation.

Another reason for a lack of generality and extensibility in existing conceptual link models might be the fact that their design is often based on a top down approach, starting from a specific hypertext domain that finally leads to a model containing details about the given hypertext domain. We propose to follow a bottom up approach where the focus is on a minimal set of linking concepts that are necessary to design a basic cross-media link and annotation metamodel that is as simple as possible but still expressive enough that it can be used to model different hypermedia domains. The resulting link and annotation metamodel can then be used to drive the development of an architecture for cross-media annotation and link services.

In the past, we have designed such a general cross-media link metamodel called the resource-selector-link (RSL) model [17]. In this paper, we show how an annotation service has been realised based on the RSL model and discuss the required architecture to support extensibility, not only on the model layer, but also on the link authoring and rendering level. The specification of such a general link and annotation architecture ensures that we no longer implement isolated linking solutions with limited support for evolution and extensibility on the authoring and information rendering layer.

3. Open cross-media annotation

In this section, we discuss the limitations of existing digital annotation tools with respect to support of cross-media annotations and introduce the requirements for true cross-media annotation tools. As mentioned earlier, we focus on annotations but our findings can also be applied to cross-media link solutions since we treat annotations as a special form of linking. Existing annotation architectures and services may be classified based on the types of resources that can be annotated as well as the potential media types that are available in annotations. To illustrate the different types of systems, we define the *annotation matrix* shown in Fig. 1. On the horizontal axis, we mark the number of different resource types that can be annotated whereas on the vertical axis we record the number of different media types that can be used in annotating a given resource.

The simplest type of annotation services, represented by the set $\mathscr{A} = \{A_1, \ldots, A_n\}$ in Fig. 1, only provides functionality for one type of resource to be annotated to be annotated with a specific type of annotations (e.g. annotate text with sound). The Annotea solution introduced earlier in this paper belongs to this category since XML documents can be annotated with textual content only. Some more flexibility is provided by systems where a single type of resource can be associated with annotations of different media types. For example, textual content can be annotated with text notes, sounds and movies. These types of annotation services $\mathscr{B} = \{B_1, \ldots, B_n\}$ are located on

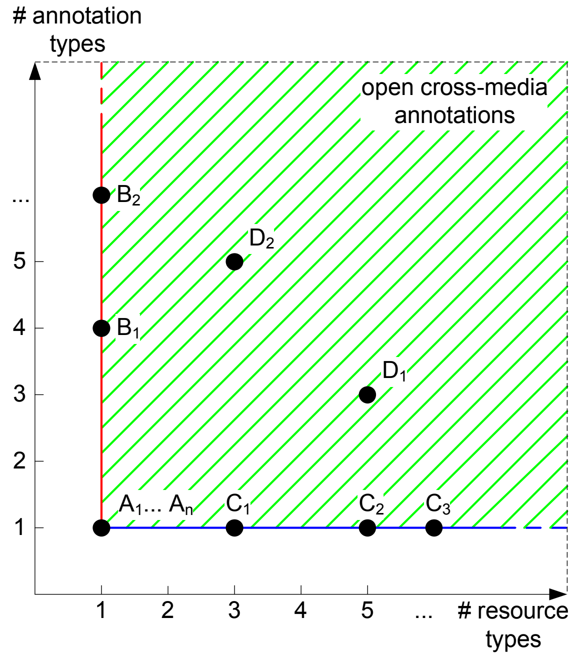

Figure 1: Annotation matrix

the vertical line going through 1. The Stickis[7] browser toolbar is such a solution where regular webpages can be annotated with a set of rich media content. A third class of systems $\mathscr{C} = \{C_1, \ldots, C_n\}$ enables the annotation of different types of resources, but with a single annotation media type only. Those solutions can be found on the horizontal line going through 1. Last but not least, we have true cross-media annotation services $\mathscr{D} = \{D_1, \ldots, D_n\}$, where a set of different resource types can be linked to annotations of different media types. An example of such an annotation service is MADCOW, where a fixed set of digital resource types—text, images and videos—can be annotated with text, images, sound or videos.

Even if we have a true cross-media annotation service, there is often a limitation in terms of there being a fixed set of media types that can be annotated and used in annotations. We aim for an extensible solution where any new type of resource or annotation can be added at a later stage. We name these types of extensible solutions *open cross-media annotation systems*. Open cross-media annotation systems are no longer represented by a single point in our annotation matrix, but rather cover the entire shaded area. While some existing solutions such as the FAST model support this kind of extensibility on the model level—at least for digital media types—we show that there is a lack of extensibility when it comes to the architecture and application or rendering level.

To illustrate what we mean by a lack of extensibility on the annotation architecture and application level, let us have a closer look at the MADCOW [6] multimedia digital annotation system. As mentioned earlier, the authoring tool for creating new multimedia annotations has been realised as a browser plug-in. The tool currently deals with text, image and video

[7] http://stickis.com

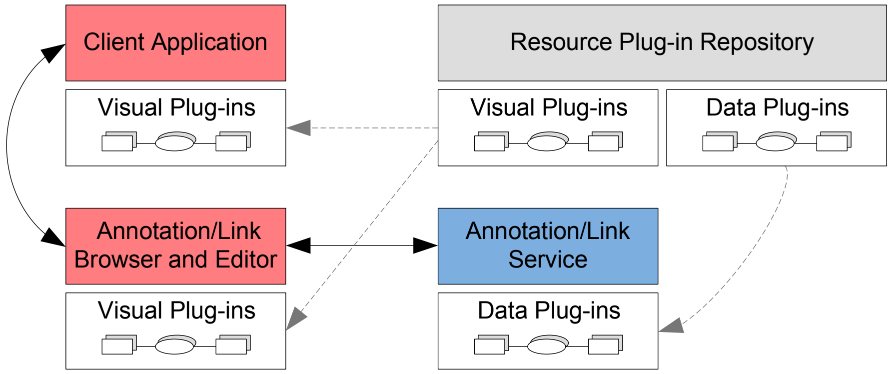

Figure 2: Open cross-media annotation architecture

annotations which is also reflected through different visual elements such as media-specific buttons in the MADCOW user interface. Let us consider what happens if somebody decides that a new media type, for example sound, should be supported by the MADCOW annotation system. Since the authoring tool has been implemented as a single monolithic component, the user interface would have to be extended to deal with the new type of resource. This implies that, for each newly introduced media type, a new version of the user interface would have to be deployed. Furthermore, since there is no flexible mechanism to dynamically extend the set of supported media types on demand, each instance of the annotation tool always has to support all existing types of resource even if some users work only with a limited subset of these media types. Last but not least, often there is not a single annotation tool but different versions, such as a browser plug-in and a standalone component, making use of the same underlying annotation model. Therefore, we have to ensure that the user interfaces of all existing annotation tools are extended individually in order to support a single new media type. This lack of flexible extensibility on the annotation tool and application level is an issue that is not only present in MADCOW, but is common to most existing annotation solutions when faced with necessity of introducing new media types. Our solution to deal with this extensibility problem is to ensure that the visual definition of annotation anchors, referred to as selectors, for a specific resource type is no longer part of the annotation authoring and rendering tool but realised in separate visual plug-in components that can be automatically installed on demand.

We propose an architecture for an open cross-media annotation system based on a cross-media annotation model that supports this form of extensibility [18]. The basic idea is that we have one or more annotation services that offer their functionality to different client applications as shown in Fig. 2. A first important thing to point out is that we make a clear distinction between the core annotation and link service and any media-specific implementation. The annotation service knows how to deal with the underlying annotation model presented in the next section, but any media-specific functionality is introduced via specific *data plug-ins*. To extend the annotation service with a new media type, a data plug-in has to be provided. An annotation service might be installed with an existing set of data plug-ins, but plug-ins can also be downloaded and installed on demand from different resource plug-in repositories (see dashed arrows in Fig. 2). Since we aim for extensibility not only on the data and model level but also on the application level, a *visual plug-in* has to be developed in addition to the data plug-in. Note that there might be different visual plug-in implementations for a single data plug-in. While it seems to be obvious to separate the media-specific creation and rendering of annotation or link anchors from the general annotation tools, it is exactly the current lack of this separation of concerns that makes it difficult to extend existing annotation and link services with new media types.

Since we do not want to force application developers to rewrite and change their entire application to make use of our annotation service, we propose a standalone annotation/link browser component that runs on the client platform. The only required communication between a client application and the annotation browser deals with information about the resource that is currently accessed within the client application. Based on a unique resource identifier, the annotation browser contacts the annotation service to get any additional external annotation and link metadata that has been defined for a given resource. The annotation browser also has to ensure that a visual plug-in for the given resource type is installed. Each visual plug-in has basically two purposes. First, it has to be able to render a specific resource type and visualise any annotation anchors that have been defined by selectors over that resource. Secondly, the visual plug-in has to provide some functionality to create and delete resources as well as selectors. After the information about the annotations has been retrieved from the annotation service, the annotation anchors will be highlighted by the visual plug-in. In the case that an annotation is selected within the annotation browser, a request is sent to the annotation server to get supplemental information for the selected annotation. As soon as another resource is accessed in the client application, the information shown in the annotation browser is automatically updated. While the communication and integration of existing applications with an annotation tool is not novel and has already been used in related approaches, the extensibility of these annotation tools is often limited due to the fact that the application logic of the tool deals with media-specific details.

Having presented the general idea of open cross-media annotation systems along with the requirements for extensibility on both the data and application levels, we will provide some details of how extensibility is achieved on each of these levels in the next two sections.

4. Annotation model

In this section, we start by looking at one of the proposed reference models for annotation services before going on to present our general cross-media annotation model that could be used as a basis for the implementation of such services.

Within the DELOS Network of Excellence on Digital Libraries[8], a reference model (DLRM) was defined to support more systematic research on digital libraries and serve as a foundation for comparing the functionality of different digital library implementations. We briefly outline the parts of the DELOS reference model dealing with annotations. This enables us to position our model in relation to the existing reference model as well as highlighting some of the major differences arising from the goal and intended use of the model.

Figure 3 shows parts of the digital library resource concept map as introduced in the DLRM document [19]. The most general concept in the reference model is the `Resource` which is used to represent any digital library entity. Particular instances of digital library resources, such as text, videos and annotations, are represented by the `Information Object` concept. A `Resource` defines some characteristics which are shared by all the different types of resources. These characteristics include a unique resource identifier, information about the format and quality as well as specific resource policy information.

The definition of composite resources is supported through the `hasPart` relation whereas the linking of different resources is enabled by the `associatedWith` relation. The annotation of arbitrary resources, or particular regions within these resources, with other information objects is represented by the `hasAnnotation` relationship between the `Resource` and `Information Object` concepts. Since we will pay special attention to the annotation mechanism while comparing our model with the reference model, we would like to give the exact definition of an annotation as provided in the DLRM document:

> An *Annotation* is any kind of super-structural *Information Object* including notes, structured comments, or links, that an *Actor* may associate with a *Region* of a *Resource* via the <*hasAnnotation*> relation, in order to add an interpretative value. An annotation must be identified by a *Resource Identifier*, be authored by an *Actor*, and may be shared with *Groups* according to *Policies* regulating it (*Resource* is <*regulatedBy*> *Policy*). An *Annotation* may relate a *Resource* to one or more other *Resources* via the appropriate <*hasAnnotation*> relationship.
>
> Candela et al. [19]

[8] http://www.delos.info

After this very brief overview of the concepts for annotating and linking resources in the DELOS digital library reference model, we now introduce our cross-media annotation model. The first thing to note is the fact that our model is defined using the OM data model [20] that integrates concepts from both entity relationship (ER) and object-oriented data models and is intended to bridge the gap between conceptual and implementation models. This means that our model can be mapped directly to database structures and is therefore a step closer to the realisation of annotation services than the typical reference models while still being at the conceptual level.

As explained in Section 2, we treat an annotation as a special type of link between two or more resources. The presented annotation model is actually an application and extension of our more general resource-selector-link (RSL) model [17] for cross-media linking as shown in Fig. 4.

The OM model supports information modelling through a separation of classification and typing. While *typing* deals with entities represented by objects with attributes, methods and triggers, the *classification* through named collections deals with the semantic roles of specific object instances. In Fig. 4, collections are represented by the rectangular shapes with the membertype specified in the shaded upper right part. The OM model provides a high-level *association* construct, represented by an oval shape, which enables associations between entities to be classified and manipulated directly. A ranking over an association is indicated by placing the association's name between two vertical lines, for example |`HasLayers`|. It is important to emphasise that OM also serves as a modelling language for a set of object-oriented data management systems, for example OMS Java [21] and more recently OMS Avon [22], and has been used to implement our link and annotation server (iServer) [23].

Similar to the `Resource` concept in the DLRM model, our annotation model introduces the generic notion of an `entity` type and all entity instances are classified and grouped by the collection `Entities`. As in the DELOS reference model, an entity has different characteristics which are shared among all specialisations of the `entity` type. Each entity is created by exactly one `individual` which is represented in the model by the `CreatedBy` association. Furthermore, access rights can be defined at the entity level by the `AccessibleTo` association. Note that these access rights can be granted on the group level or to individuals as well as to combinations of groups and individuals. A set of `contextResolver` instances can be associated with each entity which defines whether an instance is available within a specific context. Last but not least, arbitrary properties (parameters) in the form of key/value pairs can be associated with an entity by using the `HasProperties` association. This enables the extension of entities with any additional metadata required by third party applications without having to extend the core data model. To deal with complex metadata, an entity can also be associated with other entities by using the concept of a link introduced in the following paragraphs. The RSL model offers three specialisations of the abstract entity concept represented by the `resource`, `selector` and `link` subtypes.

The `resource` type represents any particular digital or physical resource that has to be managed by the annotation and link

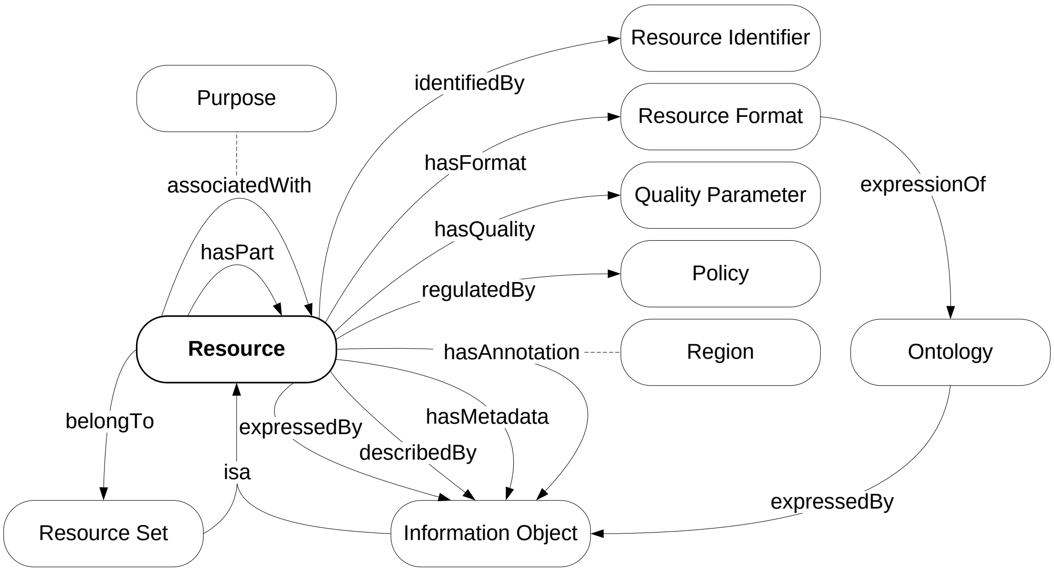

Figure 3: Digital library resource domain concept map

model. It is similar to the `Information Object` concept in the DLRM model. For each specific resource type to be supported, a new resource subtype with media-specific characteristics has to be defined via a resource plug-in mechanism.

The definition of links between different entities is supported by the `link` type. A link can have one or multiple source entities and point to one or more target entities which is reflected by the cardinality constraints on the `HasSource` and `HasTarget` associations. As mentioned earlier, we treat annotations as a special classification of links which is represented by the collection named `Annotations` in our model. Note that by treating links and annotations as first-class objects and at the same time modelling them as specialisations of the `entity` type, we gain some flexibility compared to the DLRM model where links are represented by the `associatedWith` relation. We can not only define links between resources but also create links that have other links as source or target objects. This enables us, for example, to easily add an annotation to a link; something which is not possible in the DLRM model since the `hasAnnotation` relationship cannot be defined over the `associatedWith` relation.

Often we want to link or annotate specific parts of a resource rather than entire resources. In our model, we therefore introduce the `selector` type as a third specialisation of the `entity` type. A selector is tightly coupled to a specific resource type over the `RefersTo` association and enables the selection of a specific part of a given type of resource. For example, a selector for sounds might be time-based, i.e. from time t_i to time t_j, whereas a selector for text documents could be based on character positions, i.e. from character c_i to character c_j. It is up to the developer of a new resource plug-in to not only provide an implementation for the specific `resource` type but also the corresponding `selector`. Each selector is further associated with a `layer` which, in the case of overlapping selectors, defines their precedence order. In the case that a specific selection would return several links by activating multiple overlapping selectors the link bound to the selector on the uppermost layer will be selected by definition. Furthermore, specific layers may be activated, deactivated and dynamically reordered to enable context-dependent link resolution.

How does our selector concept relate to the resource addressing functionality offered by the DLRM model? In the DLRM model, specific regions of a resource can be annotated using the `Region` concept. However, the mechanism for selecting a specific region of a resource is only available for the information object to be annotated but not for the annotation itself. This means that, in DLRM, only entire information objects can be used as annotations whereas, in our RSL-based model, also parts of resources can be used to annotate other entities. Another benefit of the selector concept and the modelling of links and annotations as first-class objects becomes evident if we revisit the concept of links provided by the `associatedWith` relation in the DLRM model. There, links can only be defined between entire resources whereas in our model we can use the selector concept to create links between specific parts of different resources.

As described earlier, our RSL-based annotation model defines any access rights at the level of entities. This has the advantage that we can not only specify if a resource is available as supported in the DLRM model by the `regulatedBy` and `Policy` concepts but also define access rights on the selector and link level. It implies that we can, for example, define that a selector which is used to annotate a resource is only available for specific users whereas the resource itself may be available for everybody. We are therefore able to specify access rights at a very fine granularity level and not just define whether an entire resource is accessible or not.

The same flexibility that has just been described for accessing annotations, links, resources and selectors based on user profiles is also applicable to the context-specific information

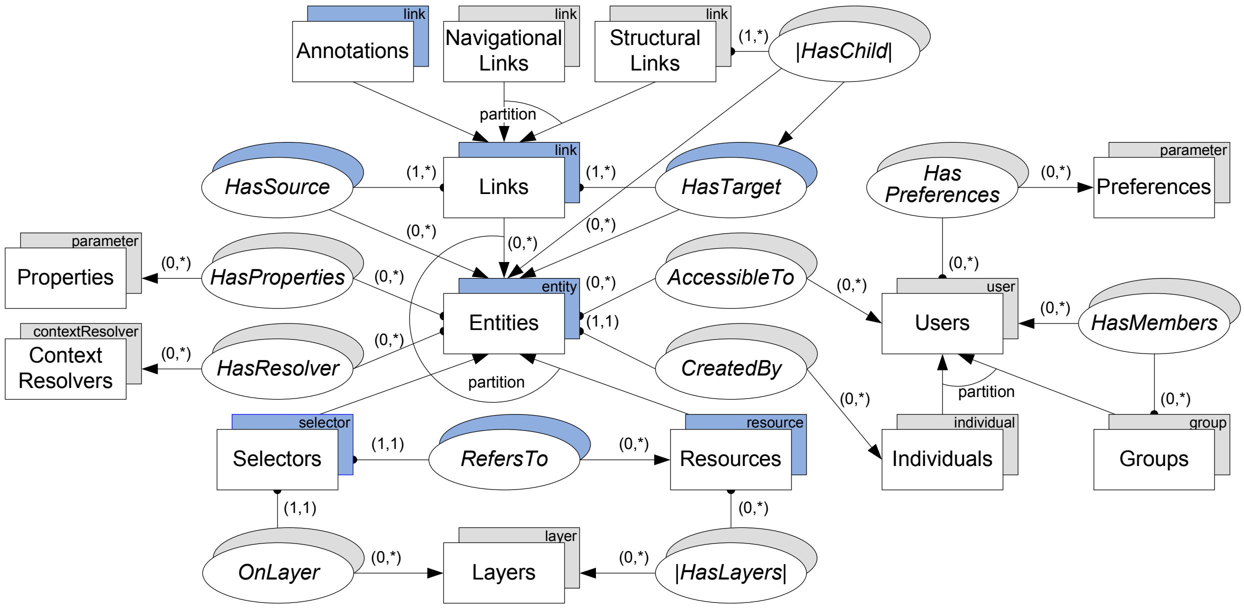

Figure 4: RSL-based annotation model

delivery based on the `contextResolver` concept introduced earlier in this section. This implies that an annotation or any other entity might only be accessible in a specific context. For example, some annotations might only become available if the user has already accessed specific resources beforehand.

A final remark has to be made about the representation of different types of annotations in our annotation model. In Fig. 4, only a single type of annotation, described by the `Annotations` collection, is shown. Of course it is easily possible to distinguish different types of annotations by introducing further subcollections. We can, for example, distinguish between formal and informal annotations as well as comments, examples and other types of annotations as illustrated in Fig. 5. Since the OM model offers the possibility that an object can be a member of different collections, it is even possible that an annotation has multiple classifications at the same time as described in [24]. Note that Annotea offers a flexible classification of annotations via the annotation subtype concept. A slightly different approach has been chosen in FAST [7], where parts of an annotation can be classified via a specific meaning mechanism.

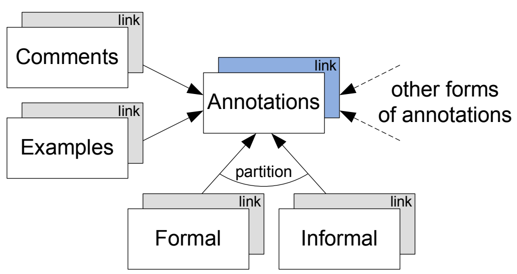

Figure 5: Classification of annotations

Our annotation model introduces some flexibility in terms of the granularity and the types of objects that can be annotated as well as used in annotations. While the model has many similarities to existing solutions, for example the DLRM model, it also shows that through generalisation and the treatment of annotations and links as first-class objects, we become more flexible in cross-annotating digital as well as physical content. While the presented model can be extended to deal with new types of media by providing specific resource and selector implementations, the management of cross-media annotation and link information is only part of the problem to be addressed. While other annotation models, such as the FAST model, also deal with media extensions at the model level, in the next section we investigate some of the problems arising when this extensibility should be supported at the application and annotation tool level. Based on our experience in implementing solutions for different types of cross-media annotations, we propose a framework for extensible annotation and browsing functionality.

5. Extensible annotation/link browser and editor

After highlighting the requirements for extensible cross-media annotation services and discussing our solution for the model layer, we now show how extensibility can be dealt with on the annotation editor and browser level. As introduced earlier, the data plug-ins are responsible for persistently storing any additional data that is required to support a new media type. In particular, a specific implementation of the resource and selector concepts have to be provided for each new data plug-in and the interface methods to create, read, update and delete (CRUD) media-specific data have to be implemented.

The functionality of a visual plug-in is defined by an interface that has to be implemented by concrete visual plug-in instances. Each visual plug-in has to provide some functionality to define new resources as well as selectors which can then be used as annotation sources or targets by the general annotation

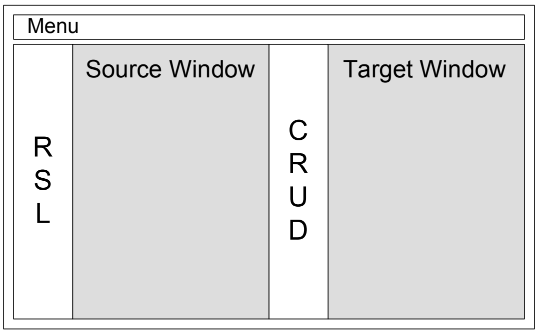

Figure 6: Annotation/link browser and editor components

tool. Furthermore, the interface defines a number of methods that are used by the general annotation tool to get access to the selector or resource that is currently selected within the visual plug-in, for example `getSelectedEntity()`. This is the only direct connection from the annotation browser introduced earlier in Fig. 2 to arbitrary visual plug-ins.

The annotation browser can not only be used to browse existing annotations but also as an authoring and editing tool to define new cross-media annotations. In the default setting, the annotation browser shows two main windows next to each other as indicated in Fig. 6. Note that these two windows always show the output of a single visual plug-in and complex documents consisting of multiple media types can currently only be visualised if this is supported by the corresponding visual plug-in. In the future, we plan to make use of the RSL model's structural link features [17] to metamodel complex composite cross-media document structures and visualise them in a generic way. The authoring tool will only have to know how to render the structural relationship between different components whereas each individual resource is rendered based on its visual plug-in.

The tool further provides functionality for the CRUD operations as well as to deal with the general RSL functionality such as the user management. To define an annotation for a given source document, the user first selects the specific part of the resource to be annotated in the source window and then annotates it with parts of the resource shown in the target window. Note that, as part of the annotation process, the user can not only select existing resources but also create new annotation resource instances based on the editing functionality offered by the visual plug-in. After selecting the `create annotation` command, the authoring tool gets access to the required selected entities via the visual plug-in interface. Since this single dependency between the authoring tool and any existing plug-ins is defined at the entity level (resources or selectors), the authoring tool does not deal with any media-specific implementation and therefore does not have to be changed at all to support a new resource type via the visual plug-in mechanism.

The default setup with two adjacent or overlapping windows for the source document and its annotation is very similar to the configuration of the Memex described by Bush, where also a source and target screen are available [25]. The major difference is that in Bush's vision there is only a single resource type, microfilm, available, whereas in our case we have a potentially unlimited number of resource types represented by the set of available data and visual plug-ins. Of course, the type of resources visualised in the two windows can be changed independently since each window is managed by a separate instance of a visual resource plug-in. Furthermore, there are different possible configurations of the annotation authoring tool with more than just the two resource windows shown in Fig. 7.

While the use of the annotation browser and authoring tool provides access to external annotation services without changing the graphical user interface of existing applications, it is also possible to integrate the visualisation functionality for specific media types directly within the client application. A client application can either make use of existing visual plug-ins or the functionality defined by the visual plug-in interface can be implemented in an application-specific manner. For example, the right-hand side of Fig. 7 shows a web browser client with a visual plug-in that we developed for the XHTML resource type. The web browser client communicates with the annotation editor and can either act as a substitute for the source or target window. The left-hand side of Fig. 7 shows an annotation browser and editor where the source and target windows are currently overlapping. The important thing to note is that each resource type is treated separately through a specific plug-in. If a user selects a highlighted annotation selector within the client, it will be checked whether a visual plug-in for the linked annotation is available and, if so, the annotation is visualised. In the case that there is no client-specific visualisation available, the annotation browser will be used as a mediator to visualise the corresponding annotation via the `showEntity()` method. This has the major advantage that we can add new types of resources to our annotation service without the client application having to know about them. Of course, if desired, the client application can then always be extended to "natively" support the new media type as shown for the web browser extension.

In the annotation authoring process described earlier, we can not only define the selectors within the authoring tool but also directly access information from the visual plug-ins installed in external client applications. In this case, the client application informs the annotation tool about the currently active selector which has to be used as an annotation source or target. This has the advantage that, for annotation-aware client applications with the corresponding visual plug-ins, any selections can be done directly within the application and only the command to create the annotation has to be issued by using the annotation authoring tool.

Various applications have been realised based on the presented cross-media annotation and link model. For that purpose, different plug-ins for digital resources such as webpages and movies as well as physical resources such as interactive paper and RFID-tagged objects have been implemented as described in the next section. While our earlier applications were based on a simpler client-server architecture, we are currently implementing the presented architecture which should finally result in the desired open cross-media annotation and link service.

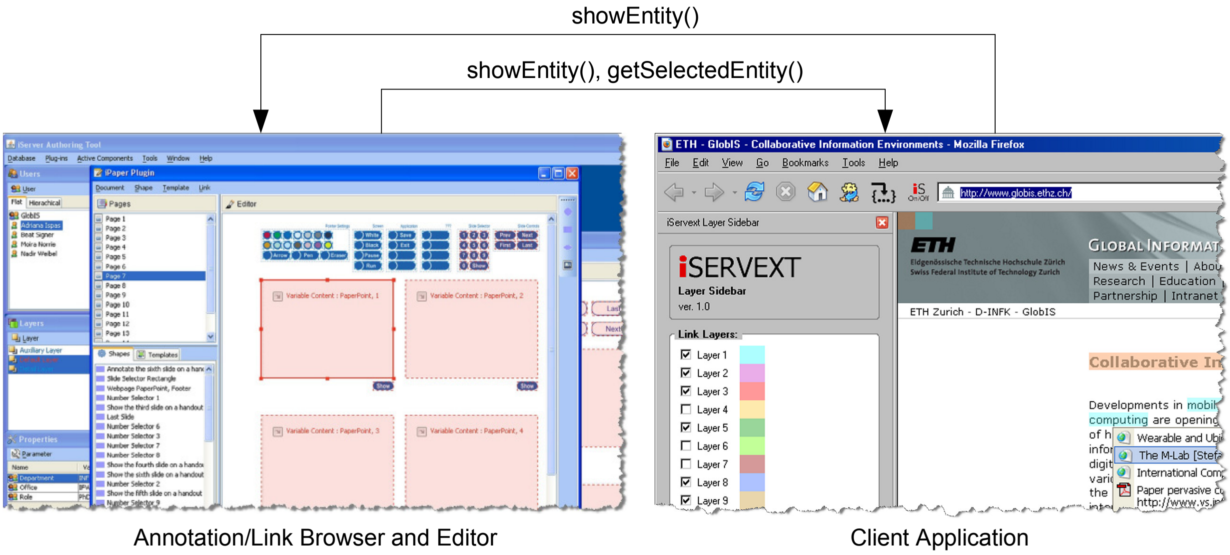

Figure 7: Annotation/link browser and editor interfacing with an external client

6. Evaluation

The evaluation of the expressiveness, flexibility and extensibility of a metamodel is not an easy task and can often only be carried out as part of a long term experiment by applying the model to implement different applications. Over the past few years, our RSL cross-media link metamodel and its extension for cross-media annotations were used in a number of applications where different types of media have been integrated via cross-media links or annotations. In the course of developing these applications, the model and architecture have been extended not only to support new media types, but also to cope with different application domains. The important fact is that the core RSL metamodel did not have to be changed to realise any of these applications and only resource-specific plug-ins had to be added. Furthermore, some domain-specific extensions were modelled by using the concepts offered by the RSL metamodel.

The RSL metamodel was implemented based on the OMS Java object database [21]. As mentioned earlier, the RSL model was mapped directly to the OMS Java database structures and extended with the required application-specific functionality resulting in a cross-media information server called iServer [23]. The iServer framework offers a plug-in mechanism where new media types can be added by implementing the resource and selector concepts introduced by the RSL model. All the concepts defined in the RSL metamodel have been implemented as part of an extension of the OMS Java object database and the iServer platform offers a Java API to create, read, update and delete information managed by the link server. In addition to the Java interface, iServer provides a language-independent XML representation of the link metadata in combination with a Web Service implementation. It is out of the scope of this paper to provide a detailed description of the iServer implementation but further details can be found in [26]. In the rest of this section, we present and comment on some of the applications that have been realised based on RSL and its iServer implementation over the last few years and outline the generality and extensibility of the RSL model as well as of our cross-media annotation and link architecture.

The cross-media link server was originally designed for augmenting paper documents with digital content and services. We therefore start by presenting the interactive paper (iPaper) plug-in as a first extension that is based on the RSL core model. The basic idea is that we have a device that enables us to track a user's position on a single page while interacting with a paper document. In the majority of our interactive paper applications, we relied on Anoto's Digital Pen and Paper functionality[9] for tracking the coordinates of a digital pen that can either be used as a capturing or pointing device. The goal was to link specific parts of a paper document to supplemental digital information and to follow these links as soon as the user interacts with the corresponding part of the document. We achieved this goal by defining a data plug-in in terms of iPaper-specific implementations for the resource and selector concepts introduced earlier while discussing the RSL metamodel. In the case of our interactive paper plug-in for iServer, a resource is represented by a document page, whereas a selector can be any arbitrary shape on the page. Since the iPaper extension was our first iServer plug-in, we were only able to link to other iPaper resources and selectors as well as some special types of resources such as HTML pages. Of course, we can not only annotate or link paper documents to existing resources but the interactive paper plug-in can also be used to create new annotations. After a digital or physical resource has been selected, any information that is written on a paper document with a digital pen can be captured to form the content of a new annotation. Since there was a need to link to other types of resources, we successively extended iServer by adding new resource plug-ins as described below. Note that the iPaper plug-in has been used to implement

[9] http://www.anoto.com

various interactive paper applications including, for example, a paper-based interactive festival guide [27] as well as an interactive paper-based proof-editing tool [28].

In another project, we extended iServer with a plug-in for HTML pages (iWeb) to link or annotate webpages in a similar way to that described earlier when we were discussing external annotation and link servers. While an iWeb resource is represented by a single XHTML page, for the definition of the selector concept, we built on concepts introduced by the XLink language. While some of the link services introduced earlier store all their link metadata in the form of XLink resources, we only use the XLink addressing scheme and store the information in iServer. However, we can map any XLink information to our iWeb representation and, if necessary, create an XLink export of information managed by the iWeb plug-in. In this way, XLink concepts that are not directly available in the RSL model can be represented without any schema modification by assigning arbitrary properties to RSL entities. Our iWeb solution offers additional features that are not available in the XLink standard, including link and resource ownership or the treatment of overlapping links via the RSL layer concept. The most important advantage is that iWeb resources can be linked to any existing or future resource type that is supported by the iServer platform. An iWeb Firefox extension was developed to directly visualise the link metadata within the web browser (see right-hand side of Fig. 7). Details about the iWeb plug-in and its browser extension can be found in [26].

Another iServer plug-in was developed to enable cross-media links and annotations for movies (iMovie). While the rendering of iMovie metadata was realised based on Apple's QuickTime development kit[10], the corresponding media-specific resource and selector extensions were addressed as part of the data plug-in. An iWeb resource is represented by a single movie where different forms of spatial and temporal movie selectors have been defined. This emphasises the fact that we are not limited to defining a single type of selector for a given resource type. The only important thing is that the type of each resource and selector is specified in a unique way to ensure that our cross-media link architecture can load the correct data and visual plug-ins.

We have already introduced the iPaper plug-in for linking paper and digital information spaces. As part of the Lost Cosmonaut art project [29] that we were involved in, the artist wanted to link not only paper documents but arbitrary physical objects to digital information and services. Similar to some existing physical hypermedia solutions, we decided to used RFID tags for the identification of physical objects which could then be used as a link source or target. However, in contrast to the often used top down approach, where the application drives the modelling process and leads to a mixed application and link model, we only had to enhance the iServer platform with an RFID plug-in. This was a relatively simple task since the corresponding RSL resource could be represented by the set of available RFID tags whereas a selector was manifested by a specific RFID identifier.

As mentioned in Section 2, the main "difference" between a link and an annotation is the fact that the annotation process often includes the creation of the linked resource. In many of our interactive paper applications, the RSL model and architecture has not only been used for authoring predefined link information but also to create new runtime annotations based on information captured by the digital pen. For example, the RSL metamodel was used to represent different types of annotations such as formal and informal annotations in [30].

During the development of some of our interactive paper and cross-media applications, it became evident that it is not sufficient to only link resources such as HTML pages or movies and that it would be helpful to directly link into application code. By introducing active components as a new form of RSL resource that contains a snippet of program code which is executed at link activation time, we became able to treat the selection of a link as an event triggered in an event-driven architecture [31]. The introduction of active components enormously simplified the prototyping and implementation of applications, since we could no longer just define cross-media links between different snippets of information, but also link to digital services. In addition, active components can be used for implementing arbitrary data transformations and act as proxy components to third-party link services. Various third-party applications and services have been integrated based on the active component approach including a paper-based image retrieval process for digital libraries [32] as well as PaperPoint [33], an interactive paper-driven presentation tool for giving PowerPoint presentations.

Our cross-media information management architecture is not limited to the classical client-server setup where multiple clients interact with a single cross-media link server. In the past, we have implemented a distributed iServer version, where different iServer instances communicate using peer-to-peer (P2P) technologies [34]. For the realisation of the distributed iServer version, the core RSL model did not have to be changed at all and it was very helpful that our cross-media model offers the concept of data ownership at the entity level. This enables a user to define which resources should be shared and which entities have to be treated as private at a very fine level of granularity and in a non-application-specific manner. More recently, we also investigated the use of the cross-media link model and architecture for new forms of personal cross-media information management [35]. Special cue resources are linked to arbitrary entities and can later be used to retrieve information from the personal cross-media information space.

By following a bottom up approach with a minimal set of linking concepts defined by the RSL model, we were able to use the same underlying cross-media link and annotation model for a variety of applications and domains. Other solutions follow a top-down approach where the model is defined for a specific application domain without a general underlying hypermedia model. This often results in problems when information from different domains should be integrated since one has to transform data between different models. As we have highlighted in Section 2, those projects that support extensibility on the model layer normally do not address the issue of extensibil-

[10]http://developer.apple.com/quicktime/download/

ity on the application and browser level. While our proposed solution does not require any changes to the annotation and link browser or editor component when a third party decides to offer support for a new media type, other solutions require major changes and extensions on the application layer. This is mainly due to the fact that there is no clear separation between the generic cross-media link and annotation functionality and the corresponding resource-specific visualisation as introduced by our solution.

As we have shown in this section, over the last few years our cross-media link architecture has been extended by a variety of new media types without having to change the core RSL model. While this proves the extensibility of our RSL model and architecture, the expressiveness and flexibility of the iServer architecture has been demonstrated by using the very same RSL model in a variety of application domains, ranging from the rapid prototyping of user interfaces based on active components, annotation sharing services, distributed cross-media information spaces as well as personal cross-media information management. Since we used the same underlying model for all of our applications, information can easily be shared and integrated across different applications without having to migrate any resources.

7. Conclusions

We have presented a model and architecture for open cross-media annotation and link services that can be dynamically extended with new media types on the model as well as on the information authoring and rendering layer. Through generalisation and the treatment of annotations and links as first-class objects, the presented RSL-based annotation and link model introduces additional flexibility in comparison to existing solutions. While a number of existing annotation and link services deal with extensibility on the model level, the same flexibility in terms of supporting new media types is often missing on the authoring and rendering layer. We have introduced an integrated open cross-media annotation solution providing a sustainable annotation and link fabric in terms of an extensible cross-media link model together with an architecture that guarantees future extensibility and ensures that annotations persist and can be reused over time. In the future, we plan to investigate the visualisation of more complex information structures where cross-media resources are no longer just associated by navigational links, but can also be composed based on the RSL model's structural link concept.

[1] B. G. Christensen, F. A. Hansen, N. O. Bouvin, Xspect: Bridging open hypermedia and xlink, in: Proceedings of WWW 2003, 12th International World Wide Web Conference, Budapest, Hungary, 2003, pp. 490–499.

[2] M.-R. Koivunen, Semantic authoring by tagging with annotea social bookmarks and topics, in: Proceedings of SAAW2006, 1st Semantic Authoring and Annotation Workshop, Athens, Greece, 2006.

[3] J. Hunter, R. Schroeter, Co-annotea: A system for tagging relationships between multiple mixed-media objects, IEEE Multimedia 15 (3) (2008) 42–53.

[4] M. Agosti, N. Ferro, A system architecture as a support to a flexible annotation service, in: Proceedings of the 6th Thematic Workshop of the EU Network of Excellence DELOS, Cagliari, Italy, 2004, pp. 147–166.

[5] D. W. Archer, L. M. L. Delcambre, F. Corubolo, L. Cassel, S. Price, U. Murthy, D. Maier, E. A. Fox, S. Murthy, J. McCall, K. Kuchibhotla, R. Suryavanshi, Superimposed information architecture for digital libraries, in: Proceedings of ECDL 2008, 12th European Conference on Research and Advanced Technology for Digital Libraries, Åarhus, Denmark, 2008, pp. 88–99.

[6] P. Bottoni, R. Civica, S. Levialdi, L. Orso, E. Panizzi, R. Trinchese, Madcow: A multimedia digital annotation system, in: Proceedings of AVI 2004, International Working Conference on Advanced Visual Interfaces, Gallipoli, Italy, 2004, pp. 55–62.

[7] M. Agosti, N. Ferro, A formal model of annotations of digital content, ACM Transactions on Information Systems (TOIS) 26 (1).

[8] S. Goose, A. Lewis, H. Davis, Ohra: Towards an open hypermedia reference architecture and a migration path for existing systems, Journal of Digital Information 1 (2).

[9] D. Millard, L. Moreau, H. Davis, S. Reich, FOHM: A fundamental open hypertext model for investigating interoperability between hypertext domains, in: Proceedings of Hypertext 2000, 11th ACM Conference on Hypertext and Hypermedia, San Antonio, USA, 2000, pp. 93–102.

[10] P. J. Nürnberg, J. J. Leggett, U. K. Wiil, An agenda for open hypermedia research, in: Proceedings of Hypertext '98, 9th ACM Conference on Hypertext and Hypermedia, Pittsburgh, USA, 1998, pp. 198–206.

[11] A. Pearl, Sun's link service: A protocol for open linking, in: Proceedings of Hypertext '89, 2nd ACM Conference on Hypertext and Hypermedia, Pittsburgh, USA, 1989, pp. 137–146.

[12] W. Hall, H. C. Davis, G. Hutchings, Rethinking Hypermedia: The Microcosm Approach, Kluwer Academic Publishers, 1996.

[13] K. M. Anderson, R. N. Taylor, J. E. J. Whitehead, Chimera: Hypermedia for heterogeneous software development environments, ACM Transactions on Information Systems 18 (3) (2000) 211–245.

[14] H. Maurer, Hyperwave: The Next Generation Web Solution, Addison Wesley, 1996.

[15] S. J. DeRose, XML linking, ACM Computing Surveys 31 (4).

[16] L. Romero, N. Correia, Hyperreal: A hypermedia model for mixed reality, in: Proceedings of Hypertext 2003, 14th ACM Conference on Hypertext and Hypermedia, Nottingham, UK, 2003, pp. 2–9.

[17] B. Signer, M. C. Norrie, As we may link: A general metamodel for hypermedia systems, in: Proceedings of ER 2007, 26th International Conference on Conceptual Modeling, Auckland, New Zealand, 2007, pp. 359–374.

[18] B. Signer, M. C. Norrie, An architecture for open cross-media annotation services, in: Proceedings of WISE 2009, 10th International Conference on Web Information Systems Engineering, Poznan, Poland, 2009, pp. 387–400.

[19] L. Candela, D. Castelli, N. Ferro, Y. Ioannidis, G. Koutrika, C. Meghini, P. Pagano, S. Ross, D. Soergel, M. Agosti, M. Dobreva, V. Katifori, H. Schuldt, The DELOS digital library reference model - foundations for digital libraries (December 2007).

[20] 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, 1993, pp. 390–401.

[21] A. Kobler, M. C. Norrie, OMS Java: A persistent object management framework, in: Java and Databases, Hermes Penton Science, 2002, pp. 46–62.

[22] M. C. Norrie, M. Grossniklaus, C. Decurtins, A. de Spindler, A. Vancea, S. Leone, Semantic Data Management for db4o, in: Proceedings of ICOODB 2008, 1st International Conference on Object Databases, Berlin, Germany, 2008.

[23] B. Signer, M. C. Norrie, A framework for cross-media information management, in: Proceedings of EuroIMSA 2005, International Conference on Internet and Multimedia Systems and Applications, Grindelwald, Switzerland, 2005, pp. 318–323.

[24] C. Decurtins, M. C. Norrie, B. Signer, Putting the gloss on paper: A framework for cross-media annotation, New Review of Hypermedia and Multimedia 9 (1) (2003) 35–57.

[25] V. Bush, As we may think, Atlantic Monthly 176 (1).

[26] B. Signer, Fundamental Concepts for Interactive Paper and Cross-Media Information Spaces, Books on Demand GmbH, 2008.

[27] B. Signer, M. Grossniklaus, M. C. Norrie, Interactive paper as a mobile client for a multi-channel web information system, World Wide Web Journal 10 (4) (2007) 529–556.

[28] N. Weibel, A. Ispas, B. Signer, M. C. Norrie, Paperproof: A paper-digital proof-editing system, in: Proceedings of CHI 2008, 26th International Conference on Human Factors in Computing Systems (Interactivity Track), Florence, Italy, 2008.

[29] A. Vogelsang, B. Signer, The lost cosmonaut: An interactive narrative environment on basis of digitally enhanced paper, in: Proceedings of VS 2005, 3rd International Conference on Virtual Storytelling, Strasbourg, France, 2005, pp. 270–279.

[30] C. Decurtins, M. C. Norrie, B. Signer, Digital annotation of printed documents, in: Proceedings of CIKM 2003, 12th International Conference on Information and Knowledge Management, New Orleans, USA, 2003, pp. 552–555.

[31] B. Signer, M. C. Norrie, Active components as a method for coupling data and services - a database-driven application development process, in: Proceedings of ICOODB 2009, International Conference on Object Databases, Zurich, Switzerland, 2009.

[32] M. Agosti, S. Beretti, G. Brettlecker, A. del Bimbo, N. Ferro, N. Fuhr, D. Keim, C.-P. Klas, T. Lidy, M. C. Norrie, P. Ranaldi, A. Rauber, H.-J. Schek, T. Schreck, H. Schuldt, B. Signer, M. Springmann, DelosDLMS - the integrated DELOS digital library management system, in: DELOS Conference on Digital Libraries, Pisa, Italy, 2007.

[33] B. Signer, M. C. Norrie, Paperpoint: A paper-based presentation and interactive paper prototyping tool, in: Proceedings of TEI 2007, First International Conference on Tangible and Embedded Interaction, Baton Rouge, USA, 2007, pp. 57–64.

[34] B. Signer, A. de Spindler, M. C. Norrie, Collaborative ad-hoc information sharing in cross-media information environments, in: Proceedings of CoopIS 2009, 17th International Conference on Cooperative Information Systems, Vilamoura, Portugal, 2009, pp. 423–430.

[35] A. Ispas, B. Signer, M. C. Norrie, An extensible framework for personal cross-media information management, in: Proceedings of Workshop on Cross-Media Information Analysis, Extraction and Management, Koblenz, Germany, 2008.