

The LEMO annotation framework: weaving multimedia annotations with the web

Bernhard Haslhofer · Wolfgang Jochum · Ross King ·
Christian Sadilek · Karin Schellner

Published online: 21 March 2009
© Springer-Verlag 2009

Abstract Cultural institutions and museums have realized that annotations contribute valuable metadata for search and retrieval, which in turn can increase the visibility of the digital items they expose via their digital library systems. By exploiting annotations created by others, visitors can discover content they would not have found otherwise, which implies that annotations must be accessible and processable for humans and machines. Currently, however, there exists no widely adopted annotation standard that goes beyond specific media types. Most institutions build their own in-house annotation solution and employ proprietary annotation models, which are not interoperable with those of other systems. As a result, annotation data are usually stored in closed data silos and visible and processable only within the scope of a certain annotation system. As the main contribution of this paper, we present the LEMO Annotation Framework. It (1) provides a uniform annotation model for multimedia contents and various types of annotations, (2) can address fragments of various content-types in a uniform, interoperable

manner, and (3) pulls annotations out of closed data silos and makes them available as interoperable, dereferencable Web resources. With the LEMO Annotation Framework annotations become part of the Web and can be processed, linked, and referenced by other services. This in turn leads to even higher visibility and increases the potential value of annotations.

Keywords Annotations · Semantics · Interoperability · Fragment identification · Multimedia · Web

1 Introduction

Digital library systems are currently in the transition from static information to dynamic knowledge spaces. While librarians and cataloguers still have the important role of creating metadata and building up a well-organized information space, the role of the visitors of digital library systems has changed: they are no longer simply passive visitors but are actively contributing and collaborating users that incorporate their knowledge into the digital library systems. An effective means that allows users to perform this task are *annotations*.

Many institutions (e.g., [31]), have realized that incorporating the end users' knowledge in terms of annotations can deliver valuable input for the cataloguing process, especially if the number of digital items to be managed is large and the available human resources for cataloguing these items are limited. Annotations can be exploited in order to search and retrieve annotated digital items [6, 19] and increase the accessibility and visibility, which is usually of interest to cultural institutions or museums.

This work makes two important contributions to this field: as a first contribution, we extensively elaborate on the subject of annotations, analyze the features of existing annotation

B. Haslhofer (✉) · W. Jochum
Department of Distributed and Multimedia Systems, University
of Vienna, Vienna, Austria
e-mail: bernhard.haslhofer@univie.ac.at

W. Jochum
e-mail: wolfgang.jochum@univie.ac.at

R. King · C. Sadilek
Digital Memory Engineering, Austrian Research Centers GmbH,
ARC, Seibersdorf, Austria
e-mail: ross.king@arcs.ac.at

C. Sadilek
e-mail: christian.sadilek@arcs.ac.at

K. Schellner
Vienna, Austria
e-mail: karin.schellner@gmx.net

solutions against a requirements framework that has been derived from state-of-the-art literature, and identify a set of novel needs which, we believe, future annotation solutions must meet in order to integrate with Web-based environments.

Since we believe that these needs have an impact on the design of novel annotation approaches in the digital libraries domain, our second contribution concerns our LEMO Annotation Framework, which specifically addresses these requirements by providing a uniform, standards-based multimedia annotation model for various content-types, can address media fragments in a uniform way, and makes annotations available as dereferencable Web-resources that can be exploited by other applications and services in order to increase the visibility and accessibility of the exposed contents. With the LEMO Annotation Framework we aim to provide a foundation for Web-based annotation tools that go beyond existing solutions. The main properties of the LEMO framework are:

- **Linkable:** annotations are first-class objects identified by their HTTP URLs, which allows external applications to link to existing LEMO annotations by referencing their URLs.
- **Extensible:** the annotation framework is extensible in order to support various annotation-types and upcoming paradigms such as tagging or structured annotations. Furthermore, it is possible to add support for specific content-types (e.g., AVI, PDF) without redesigning and rewriting existing system components.
- **Multimedia-enabled:** annotations can address digital items of any content-type and take into account content-type specific characteristics. Hence, LEMO supports a *uniform annotation model* and *uniform fragment identification*.
- **Open and Interoperable:** annotations are published on the Web and can be accessed by any other external application unless they are protected for legal reasons. They follow existing standards and are therefore read- and interpretable by other applications that are aware of these standards.

The requirements driving the LEMO approach are discussed in detail in Sect. 3, but are summarized as follows: First, annotation systems should consider that the digital items exposed by modern digital library systems (e.g., [35, 47, 55, 60]) are multimedia, i.e., audio, video, image, etc. This requires a *uniform annotation model* instead of isolated, content-type specific solutions. Furthermore, novel paradigms such as tagging [34, 61] and collaborative filtering [33], which are tightly related to the concept of annotations, should be considered in such a uniform annotation approach.

Second, there is a need for uniform methods to address specific content parts or regions in digital items to be annotated. This could be, for instance, certain paragraphs in documents, frames in videos, or areas in images [30, 57]. Considering the various possible types of multimedia content, this calls for *uniform fragment identification*, i.e., a strategy to reference fragments in various content-types in a uniform, media-format independent, and interoperable manner.

Third, we can observe a shift towards the Web: digital library systems are no longer isolated, monolithic databases but have started to expose their digital items on the Web and link them with items in other library systems or resources on the Web (e.g., [36]). We believe that for annotations this shift is necessary as well: they should become open Web resources that are linkable and dereferencable also from outside the scope of a certain digital library system. If other external applications can access and process the exposed annotation data, this will further increase the visibility of the annotated digital items.

Two annotation tools have already been developed on-top of the LEMO Annotation Framework: one operates in conjunction with the FEDORA digital library system and provides a Flash-based Web interface for annotating images and videos. The other is being developed for TEL¹ and also supports the annotations of images and videos. The current releases of the annotation tools being developed for TEL can be accessed online at: <http://dme.arcs.ac.at/image-annotation-frontend> and <http://dme.arcs.ac.at/video-annotation-frontend>.

This paper is structured as follows: Sect. 2 gives an introduction to annotations in the digital library domain, derives a set of requirements from the relevant literature, and evaluates existing annotation tools against these requirements. Thereafter, in Sect. 3 we discuss additional requirements that should, in our opinion, be supported by next generation annotation tools. In Sect. 4, we describe the details of the LEMO Annotation Framework and how it supports these requirements. To demonstrate the practical feasibility of our approach we present two annotation tools that have been built on-top of LEMO in Sect. 5. After a discussion on the proof of concept in Sect. 6, we conclude this paper with Sect. 7.

2 Background and related work

In this section, we give an introduction to annotations as they are conceived in the digital libraries domain. Then we derive a set of standard requirements existing annotation solutions should fulfill according to state-of-the-art annotation

¹ The European Library TEL: <http://search.theeuropeanlibrary.org/portal/en/index.html>.

literature. Thereafter, we analyze a representative set of annotation tools against these requirements.

2.1 Annotating online cultural assets

There has been a great deal of research in the domain of annotations in the digital world [4, 8, 14, 38, 39]. Annotations often differ in their definition depending on the domain in which they are applied. An annotation can be seen as a remark, explanation or interpretation added to the original document. It is a means to make *implicit* structures *explicit* [18] and provides additional meaning to the document or passage it refers to. Ovsiannikov et al. [46] define an annotation as *a datum created and added by a third party that can take the form of a written note, a symbol, a drawing or a multimedia clip*.

Many studies have been performed in order to understand and analyze the different kinds of annotations, their use and the environment and workflows in which they are created [18, 41, 46]. Annotations can take different forms and function as analyzed thoroughly by Marshall [40]. Marshall differentiates between *formal* and *informal* annotations, whereby a formal annotation is described as metadata that follows a structural standard. Informal annotations are unstructured and therefore support only limited interoperability. The form of an annotation is furthermore divided into *implicit* and *explicit*; while an explicit annotation allows others to interpret it and is therefore also intended for sharing, implicit annotations are often only interpretable by and useable for the original annotator. Marshall makes further divisions: concerning the function of annotations she sees the dimensions of *annotation as writing vs. annotation as reading*, *extensive vs. intensive* annotation, *permanent vs. transient*. She defines two more dimensions that are concerned with the exchange of annotations: *published vs. private* and *institutional vs. workgroup vs. individual*.

Annotation capabilities and the possibility to freely and easily organize and categorize the physical documents on their desk are among the most essential reasons why people still tend to print out documents and read them in paper form. Annotations in books or other printed documents have a long tradition and their added value to both the creator and potential readers is evident. Today we can notice the trend to offer annotations also for digital content. But the variability of form and function of annotations as described above—also depending on the context and domain they are used in—remains a significant challenge when planning to transfer annotation workflows into the digital world. Many collaborative websites and community portals have discovered the added value of annotations and are already offering annotation tools in varying quality. When concentrating on the environment and domain of Cultural Heritage and Digital Libraries, a thorough approach is needed that best can

deal with the variability of annotations' form and function and the different requirements raised from different content-types and moreover considers reuse, sustainability and preservation of digital annotations.

Scientific approaches such as those followed in the MAD-COW [11], IPSA [2], and Collate [12] projects, already address some of the aspects needed for a more generic system with better support for multiple content and annotation types. With the Digital Library Annotation Service (DiLAS) project, Agosti et al. [3] focus on the design and development of an architecture and framework for managing annotations independently from a specific digital library management system. Phelps and Wilensky [51] present the idea of a *Multivalent Annotation Model* and its implementation in form of their *Multivalent Browser*. They envision a multi-layer approach: each document consists of several layers, annotations forming one of them. Any document that has a media adapter for the multivalent browser can be shown and annotated. Currently supported formats include: HTML, PDF, and TeX DVI pages.

The question whether annotations are content, metadata, or even dialogue acts has often been discussed within and between communities (e.g., [7, 8]). In the context of our work, however, we consider them as metadata and rely on interoperability strategies that have been developed for solving problems connected with metadata heterogeneities.

2.2 State-of-the-art requirements

Most of the requirements which we are describing here have been identified by Marshall et al. [3, 21, 39] and have in the past been considered as very important for annotation systems to be useful and accepted by the user communities.

Some of the requirements affect the underlying annotation model, some address the graphical user interface of the annotation tool, and others the environment in which the tool is embedded.

1. *Different content-types*. An important criterion for annotation systems is the content-types (e.g., image, audio) and formats (e.g., JPEG, MP3) for which they are designed. While some annotation tools support annotations only for a certain content-type, others can annotate various kind of digital objects, such as images, videos, documents, audio samples, etc.
2. *Segment-based annotations*. In the traditional annotation workflow, people are used to select portions of text or to mark regions in images, so the annotation model of a digital annotation system must provide a concept for modeling media parts and their interrelation, i.e., an annotation system should support segment-based annotations. The annotation tool should allow people to select joint and disjoint text passages, regions

in images, frames in video files, or sample sequences in audio files.

3. *Associations*. An important aspect of annotation systems is the ability to bring different documents into relation and to support associations between documents or parts of documents. Examples of this can be found in the domain of comparative literature where text passages of different works are analyzed and compared with each other. The concept of hyperlinks is one possible cross-reference method for documents. In an enhanced form, hyperlinks provide the possibility to assign types (e.g., “is-parody-of”) or free-text notes to associations.
4. *Reply threading*. To support users in their collaborative work, an annotation system should provide the possibility for annotation threads, i.e., nested annotations should allow users to discuss on a certain topic or subject. Annotation threads must be considered when designing the annotation model as well as in the design of the annotation application’s user interface.
5. *Controlled vocabulary support*. Annotations often occur in the form of free text without structure. When, however, annotations should support the collaboration between experts, it is essential to integrate controlled vocabularies such as taxonomies or thesauri to ensure a common understanding of the domain and support semantic search. When predefined vocabularies are applied, the annotation level is considered to be *controlled* whereas if they are not, the annotation level is considered to be *free*. Annotation systems might offer free text annotations, controlled annotations or both.
6. *Robust positioning*. As documents may undergo changes, one must consider what effects this can have on the annotations referring to the document or a part of the document. Annotations could become completely useless or even wrong if a document changes or parts of a document are deleted. Therefore it is important to find a strategy for annotations when the source documents change. One solution could be the versioning of documents and annotations, another could be the approach of robust positioning [13,50,51] which should guarantee that annotations are robust enough to survive at least modest document modifications. In [50] Wilensky and Phelps describe an algorithm for robust positioning of annotations on documents that uses unique identifiers, tree walk descriptors and context descriptors.
7. *Semantic interoperability*. Whenever applicable, annotation systems should use standards for storing annotations or should at least provide export and import of annotation data that follow a particular standard. The W3C Multimedia Semantics Incubator Group² has analyzed the current status and future requirements for enforcing and supporting annotations on the Semantic Web. They concluded, that Semantic Web technologies are practical tools for media annotations on the Web, but commonly accepted and widely used vocabularies for annotations and standards to address subregions in digital items are still missing.
8. *Collaborative*. Individual annotations are created by a particular person and are intended for later use by this person, i.e., for recalling important aspects of some document, obtaining a quick overview, etc. However, many use cases in the domain of annotations only make sense when considering the act of annotating as a collaborative task. Sharing annotations and working on annotations in a collaborative manner opens many more possibilities and is predestined to support user groups and communities in their work. However, as stated under the next point, annotations that are intended for the public may differ in form and content from annotations that are intended for private use only.
9. *Public and personal annotations*. In [41] Marshall and Brush carried out a thorough analysis on personal versus public annotations. Their findings revealed that people tend to annotate very differently if their annotations are intended only for personal use. Annotations for online discussions are differing in both form and content. If a personal annotation is being shared, it usually undergoes dramatic changes in order to make it intelligible to others. An annotation system should support both kinds of annotations, ideally with the possibility of transferring personal annotations to public annotations.
10. *Fine-grained access control*. Controlling access to annotations is another requirement in collaborative annotation systems. It is essential to allow users to control access to their contributions, e.g., to make a distinction between users that have simple access (read access) and users that have full access (read and write).
11. *In situ representation*. A progressive user-interface should allow the users to make their annotations directly on the document or the part of the document where they refer to. Nevertheless, the original document should remain readable and the annotations should be made well distinguishable from the source document.
12. *Searchable*. Annotations reveal their real power and added value when they are stored in a way so that they are easily searchable and retrievable. The system can support free-text, structured, and faceted search. Besides searching for the annotations themselves, annotations can moreover be used for formulating queries over digital items and for retrieving the most relevant ones for a query. Agosti et al. [5]. are showing *how annotations can be exploited as a useful context in order to retrieve documents relevant for a user’s query*.

² W3C Multimedia Semantics Incubator Group: <http://www.w3.org/2005/Incubator/mmsem/>.

Frommholz et al. [20] also discuss how annotations can be a helpful means for the retrieval of documents in digital library systems.

13. *Annotation management area.* The annotation tool built on top of the annotation system should provide a smart annotation management including the personal organization of annotations and an intelligible search and filter mechanism. It should be considered that studies have shown that the presentation of annotations are most useful to the users when shown in their context.
14. *Web application.* Depending on the user community addressed, the advantages and disadvantages of stand-alone versus Web applications should be considered. If the community is distributed or very heterogeneous, a Web application will be the better choice. This brings limitations in the implementation of some functionalities but these can partly be resolved by the new possibilities of emerging Web 2.0 technologies. Web applications we can further divide into those using HTML and JavaScript only, and those using other plugins (e.g., Flash, Java Applets).

2.3 Analysis of existing tools

In order to evaluate the state-of-the-art with respect to the standard requirements we have listed in the previous section, we have selected an incomplete but representative sample of annotation tools and systems that cover scientific and commercial approaches and include desktop as well as Web applications. The comparison tables in Fig. 1 indicate that many tools or systems cover only a portion of the requirements established in the previous section and that most of them focus on a specific content-type. Only approaches like MADCOW [11], Vannotea [26] or Multivalent annotations [51] are considering support of multiple content-types. Segment-based annotations are supported by almost all tools but often only for one particular content-type, e.g., selectors for images (Fotonotes™[56], Flickr [62], Photostuff [42], Zoomify Annotation System [63], etc.). Adobe Acrobat Professional [1] or the PDF Annotator [24] are also clearly concentrating on only one content format. Requirements such as supporting associations, reply-threading and taxonomies/ontologies, which we grouped under *flexible annotation types*, are mainly addressed by tools and prototypes that have their origins in the scientific world (such as Vannotea, MADCOW, Debora [44]). This may stem from the fact that these concepts are difficult to communicate to users and even more difficult to integrate into an easy-to-use tool interface.

2.4 Observations

The underlying annotation systems of our test candidates vary substantially. One-third of them are stand-alone

applications while the others were designed for the Web. The general trend points into the direction of Web applications (e.g., Flickr, Viddler, Google Notebook [23], Mojiti [43], etc.), although XLibris [53] offers a smart stand-alone approach with a high-resolution pen tablet display. Only a few of the annotation systems offer some kind of annotation management in varying elaboration levels (i.e., threading). Almost all allow collaboration by sharing annotations, but only some have already realized the importance of private and public annotations and only a few allow fine grained access control in the form of read-write-execute-permissions for different users or user groups. Only three tools (Annotator [46], Yawas [17], and Multivalent Annotations [51]) are concerned with changes in the original document and their effects on attached annotations. All three use *robust positioning* to cope with this problem. By contrast, almost all analyzed tools provide some kind of *in situ representation* of their annotations, while several ones realize the need for a searchable and filterable annotation information space.

Finally, there were only a few systems (Annotea [32], PhotoStuff [42], Vannotea [26], and M-Ontomat-Annotizer [48]) concerned with interoperability, i.e., the use of standards for the annotation schema and/or the possibility of exchanging annotation information.

During our analysis we have observed that in the case of annotation tools that have been implemented as add-ons for existing collaboration systems (like Flickr [62] or Viddler [58]) there were generally fewer supported features, as the focus was apparently more on the user-interaction with the associated media files. The previously mentioned scientific approaches often offer more challenging features, sometimes leading to a corresponding reduction in the usability of the tools.

Our analysis revealed that among all annotation tools under consideration, Vannotea fulfills most requirements that have been derived from the state-of-the-art literature. It is, however, implemented as a standalone desktop solution and can hardly be integrated into Web based environments, such as library portals.

Considering the fact that the number of annotation systems is growing, these systems should support the creation of annotations that can easily be re-used, migrated, utilized for searching across various annotation platform, or even dynamically aggregated in mashups.³ In systems that support multimedia digital items, a uniform approach for supporting multiple content- and annotation-types needs to be established. In this context, we have identified new requirements such as having a uniform annotation model and a method to provide uniform fragment identification. This will separate annotation and content-type specific characteristics and

³ A mashup is an application that combines data from several external sources.

Legend: ● Yes ○ No ⊙ Partly n/a not applicable ? unknown

Requirements/Tools and Applications	Adobe Acrobat Prof	Annotator	Annotea (Amaya/Annozilla)	BRICKS Annotation Tool	Deborra	Flickr	Fotonotes™	Google Notebook	PDF Annotator
Different content-types	○ ¹	○ ²	○ ²	○ ³	○ ³	○ ³	○ ³	○ ²	○ ¹
Segment-based annotations	●	● ⁴	●	●	●	●	●	●	●
Associations	○	○	○	○	●	○	○	○	○
Reply threading	●	○	●	●	○	○	○	○	○
Controlled vocabulary support	○	○	○ ⁵	○	○	○	○	○	○
Robust positioning	n/a	●	○	○	○	○	○	○ ⁶	n/a
Semantic interoperability	?	○	●	○	○	○	○	○	○
Collaborative	●	●	●	●	●	●	●	●	● ⁷
Public and personal annotations	?	●	●	○	●	○ ⁸	○	○ ⁹	○
Fine-grained access control	○	○	○	○	○	○	○	○	○
In-situ representation	●	⊙ ¹⁰	●	●	●	●	●	○	●
Searchable	●	●	○	●	●	○	○	●	○
Annotation management area	●	?	●	○	●	○	○	●	○
Web application	●	●	● ¹¹	●	○ ¹²	●	●	● ¹¹	○

- ¹ only PDF documents
- ² only webpages
- ³ only images
- ⁴ only when highlighting
- ⁵ predefined annotation types and possibility for new (uncontrolled) types
- ⁶ only jumps to specified web address not to specified clipping
- ⁷ restricted: editable only with PDF Annotator product
- ⁸ only whole photos can be set public/private
- ⁹ sharing only on Notebook level not on individual annotation level
- ¹⁰ only when highlighting; notes are for the whole document
- ¹¹ browser extension
- ¹² Java Applet

Legend: ● Yes ○ No ⊙ Partly n/a not applicable ? unknown

Requirements/Tools and Applications	PhotoStuff	MADCOW	Mojiti	Mc-Ontomat- Annotizer	Multivalent Annotations	Vannota	Viddler	Xlibris	Yawas	Zoomify Annotation System
Different content-types	○ ¹	●	○ ²	●	● ³	● ⁴	○ ²	⊙	○ ⁵	○ ¹
Segment-based annotations	●	●	●	●	●	●	●	●	●	●
Associations	○	●	○	○	○	●	○	?	○	○
Reply threading	○	●	○	○	● ⁶	●	○	○	○	○
Controlled vocabulary support	●	○ ⁷	○	●	○	●	○	?	○ ⁸	○
Robust positioning	○	○	○	○	●	?	○	?	●	○
Semantic interoperability	●	○	●	●	○	●	○	○	○	○
Collaborative	●	●	●	●	●	●	●	?	○	●
Public and personal annotations	○	●	●	○	○ ⁹	●	○	○	●	○
Fine-grained access control	○	●	○	○	○	●	○	?	○	○
In-situ representation	●	● ¹⁰	●	●	●	●	●	●	●	●
Searchable	●	●	●	●	○	●	○	●	●	○
Annotation management area	○	○	●	○	○	●	○	●	●	○
Web application	○	● ¹¹	● ¹²	○	○	○	● ¹²	○	●	● ¹²

- ¹ only images
- ² only video
- ³ generic approach: media adapters allow to integrate various document formats; mainly for text and images; no appropriate annotation type (frame/sequence) for audio and video files!
- ⁴ .mov, .mp4, .mp3, .qt, .mpg, .mpeg, .wmv, .avi, .pdb, .mol, .xyz, .cml, .dcm, .jp2, .jpg, .tif, .bmp, .jpeg, .giv, .flv, .html
- ⁵ only webpages
- ⁶ very restricted: Notes are said to be annotable
- ⁷ only labels for the type of annotations
- ⁸ only predefined document types/topics
- ⁹ all public
- ¹⁰ but only as symbol
- ¹¹ browser extension
- ¹² Flash

Fig. 1 Annotation tools comparison table

guarantee better interoperability and re-use. Furthermore, the architecture of such an annotation system must build on widespread technologies, which will ease the integration process in existing systems. These requirements go beyond the current state of the art and are described in the following section.

3 Requirements beyond the state of the art

In this section, we discuss additional functional requirements for Web-based annotation tools, which we consider as important in order to meet recent developments in the Digital Library and the Web domain. These requirements have been the main motivation for developing the LEMO Annotation Framework.

3.1 Uniform annotation model

In the context of LEMO, annotations are information items that follow a certain structure, have a specific semantics, and are part of a digital library system's information space. The 5s model proposed by Gonçalves et al. [22] provides a first, formal abstraction of a digital library information space and builds the basis for the digital library reference model, proposed by [15], which introduces the notion of annotations as first-class objects. Agosti et al. [7] have further formalized the main annotation concepts and defined them as *digital objects* within a digital library's information space. In their conception *an annotation must annotate one and only one digital object, which can be a document (a multimedia content item) or another annotation, i.e., an annotation must have one and only one annotation link to another digital object.*⁴

LEMO must take these well-established concepts into account and provide a uniform annotation model that offers flexibility along two dimensions: the *content-type* and the *annotation-type* support.

It is obvious that annotations for distinct content-types require different models that will share only a limited number of common elements. A video-annotation, for instance, requires time-based elements for addressing a series of frames in a certain video, while for text-annotations other elements such as `paragraph` or `lineNumber` are relevant. At the same time, there are elements such as `author` or `label` that are independent of any content-type. By strictly separating fragment identification from the basic core annotation model that contains all content- and annotation-type independent elements, LEMO provides a solid base model that can be easily extended to different content and annotation types.

Possible annotation types are free-text annotations, tags, or structured annotations. Free-text annotation is self-

explanatory—the user annotates a digital item with some freely-chosen text. Tags are controlled by users and user-communities and allow them to annotate digital items with a weak form of controlled vocabulary. Structured annotations are mainly contributed by expert users who have detailed domain knowledge and an interest in precise semantic definitions and the quality of data they produce. Since controlled vocabularies such as the Dewey Decimal Classification System (DDC) [45], or the Library of Congress Subject Headings (LCSH) [37] play an important role in organizing a digital library's information space, they are also an important part of structured annotations. Annotation-types can also include additional features, such as giving the user the possibility to reply on annotations created by other users or to relate digital items by means of annotations.

The goal of the LEMO Annotation Framework is to provide an annotation model that unifies these two dimensions in a single extensible annotation model.

3.2 Uniform fragment identification

Annotations often refer to specific parts of a digital item. They could, for instance, address a certain region in an image, or a specific sequence of frames in a video resource. In order to fulfill this basic requirement, the annotation architecture must provide means to select distinct parts or *fragments*⁵ of a digital item; at best independent of its content-type.

Besides common requirements like robust positioning, presentation control, and expressiveness [30], *interoperability* is the most important requirement of uniform fragment identification in multimedia annotation systems. A simple unified method to specify fragments is needed and critical for the targeted adaptability of the system.

Fragment definitions can have different forms that vary greatly, depending on content- and annotation-type. In order to create a system that allows for easy integration of several content and annotation types it is preferable to have content and annotations, as well as their models, clearly separated and reusable. The definition of fragments can either be part of the content, part of the annotation, or part of the link that associates the annotation with the digital item [21].

HTML is an example of a resource format that allows internal specification of fragments within the content and accordingly the resource format. Elements can have given names that can be used as link targets. However, such an internal definition limits the range of fragments that can be addressed, i.e., annotated, to those defined by the author of

⁴ In this paper we use the term *digital item* instead of *digital object*.

⁵ We use the term *fragment* to refer to any part of a digital item. Although a fragment is characterized as a part broken off or something that is small or even insignificant we choose this term in favor of others like *segment*, *part*, *piece*, *portion*, *element*, or *component* because it is neutral in relation to the origin of the part that is addressed.

the resource. Without modification of the resource, which requires write access to the resource, there is no way to add new fragment definitions (in case of HTML—anchors) to existing resources. An annotation system following this approach would be severely constrained.

External definitions shift the problem of identifying a specific fragment within a digital item away from the item's content format to external places like the metadata format or the resource identifier. This enables the definition of media fragments without the need to modify the original resource. Since the definition of the fragment's location or area is then separated from the representation of a digital item, problems like misplaced or dead target locations can occur if the digital item is modified, moved or deleted.

Metadata formats like the *Multimedia Description Scheme* (MDS) of MPEG-7 [52] or the `area` element of METS⁶ are examples how addressing of specific fragments of a media object can be integrated into a metadata format. While this is a suitable solution for a set of limited resource formats, it becomes impractical for a system that supports a larger number of media formats. Aside from the drawback that the metadata schema is growing and dependent on the number and types of media formats, the integration of the fragment definition within the metadata format becomes an obstacle, when the fragment definition needs to be exchanged. Most formats, such as the previously mentioned MPEG-7 MDS and the METS `area` element, differ largely in semantic expressiveness and syntactic expression.

By shifting the problem to the level of the identifier, which is used to link an annotation with the digital item, the fragment definition can be separated from annotation-specific formats and only depends on the format of the digital item itself. While the annotation or metadata format may vary depending on the usage scenario, architecture, or meta-format decisions, the identifier format will be consistent across various annotation systems. We think that this separation is essential in an interoperable annotation system that needs to support various content and annotation types.

We can summarize this requirement as follows: a unified way to address fragments within digital items needs to be separated from the item itself. In order to support content- and annotation-type independence, fragment identifiers should be part of the link between an annotation and the annotated digital item.

3.3 Integration with the web architecture

Since the LEMO Annotation Framework should provide the basis for Web-based annotation tools, we need to integrate it with the Web architecture [29] and treat annotations as

machine- and human-interpretable resources that can be dereferenced via their URIs. This allows client applications residing outside the system boundaries of a certain digital library system to exploit these annotations for search and retrieval tasks, which in turn increases the visibility of the digital items provided by a certain digital library system.

The recently started W3C linking open data (LOD) community project⁷ provides a set of guidelines for publishing and interlinking data on the Web and has already implemented them for a variety of data sources. We believe that one should follow the same strategy for annotation data and make them available as open data on the Web. To do so we have adapted the so-called linked-data principles [10] to the context of Web-based annotations. They demand that:

1. Annotations and the annotated digital items must have URIs as names.
2. Those URIs must be HTTP URLs so that people can look them up.
3. When a human using a Web browser or an application looks up an annotation URI, it must provide useful information, i.e., interpretable annotation data for humans and machines.
4. Annotations should include links to related resources, so that one can discover more things, i.e., the annotated digital items or other related annotations.

4 The LEMO annotation framework

After having discussed three main requirements that go beyond the state of the art in the domain of annotations, we now describe how these requirements find their technical manifestation in the LEMO Annotation Framework. First, we give an overview of its basic architecture and continue with the core of LEMO, which is a uniform, multimedia-enabled annotation model. Then we describe how we address the problems of fragment identification in a uniform, interoperable manner. Finally, we present how the annotations managed by the LEMO framework are exposed on the Web as dereferencable resources and how they can be accessed by external clients or applications.

4.1 Basic architecture

A considerable number of annotation systems investigated during the writing of this paper is related to or even build on the Annotea specification [32]. Vannota, for instance, which according to our analysis is one of the outstanding annotation systems, is based on the Annotea architecture. In principle, our annotation architecture builds also on the

⁶ Metadata Encoding & Transmission Standard <http://www.loc.gov/standards/mets/>.

⁷ The LOD project: <http://esw.w3.org/topic/SweoIG/TaskForces/CommunityProjects/LinkingOpenData>.

design of Annotea because of its simplicity and Web-orientation. Nevertheless, in order to meet the previously described requirements, we had to extend the Annotea architecture.

Regarding the original Annotea system, one of the most important architectural detail we have adopted, is the annotation representation format, which is based on RDF. Next, the LEMO architecture retains the concept that all annotations are kept in a separate repository, which is remotely accessible via a simple HTTP interface. Annotations are retrievable Web resources and identified via their associated URI. Thereby the LEMO Annotation Framework becomes an independent, separate service residing adjacent to existing digital library systems. This brings two main benefits: first, it is not necessary to break up existing structures for adding annotation behavior to digital libraries. Second, the user-contributed annotations are kept separately from bibliographic metadata, which is necessary because annotations are not per se verified by the institutions.

Besides the existing Annotea interface, LEMO provides an additional REST⁸ interface, which supports the basic CRUD (create, read, update, delete) operations on annotation data.

In addition to the simple query interface of Annotea, the system provides a SPARQL query interface for selective access to stored annotations. With this *simplicity-first* approach, we can easily implement an annotation repository without relying on heavy-weight alternatives, such as Web Services. Annotations are stored using the HTTP POST operation, HTTP GET is used for annotation retrieval and executing SPARQL queries, HTTP PUT updates annotations, and HTTP DELETE removes annotations from the repository. The annotation repository further includes a full-text search engine which indexes all incoming annotations.

The advantage of the REST-style approach is that annotations, like the digital items they annotate, become Web resources themselves; as a consequence, they can be dereferenced via their URI.

4.2 The LEMO annotation model

Our solution for providing an interoperable, multimedia-enabled annotation model is called *annotation profiles* and is derived from the concept of *application profiles* (see, e.g., [9, 25]), which is a well-known interoperability strategy in metadata concerned communities. Annotation profiles allow the definition of content- and annotation-type specific model extensions, while providing a high degree of interoperability with agreed-upon annotation standards.

One of our main goals is to achieve interoperability not only among annotations of different content- and annotation-

types but also with other system. Reusing existing vocabulary and schema definitions is one of the main rules to obey in order to achieve interoperability on a semantic level. Therefore, we semantically link the LEMO core schema elements with existing vocabularies, such as the Annotea annotation schema,⁹ which in turn reuses part of the Dublin Core Element Set [16].

Annotea has already defined a small set of model elements that reflect an annotation and has semantically linked some of these elements to Dublin Core elements. We believe that, besides minor modifications, this approach perfectly suits the needs of our core annotation schema. As illustrated in Fig. 2, the annotation core schema refines the original Annotea schema: it defines a class *Annotation* and a set of properties: *annotates*, *author*, *label*, *created*, *modified*, and *fragment*. All elements are defined in OWL and are semantically linked with the Annotea schema elements.

Extensions of the LEMO core schema can easily be created by defining an OWL ontology having a unique namespace, which should be a resolvable URI, and creating subclasses and sub-properties of the defined model fragments. Figure 3 shows two possible extensions: one enables the system to define different resources in relation to one another (Annotation Relationship Schema), and another one enabling textual annotations (Text Annotation Schema).

The LEMO core schema can easily be refined and extended by means of add-ons that define their own, content- or application-type specific annotation profile. The reuse of existing schema element definitions is the main goal of add-ons. It is possible to define dependencies among add-ons so that one add-on can reuse all the artifacts provided by other add-ons: their model artifacts, their view components, and their functionality. With that approach it is possible, for instance, to define a generic add-on for the content-type *image* and extend it by lightweight content-type add-ons for specific image formats (e.g., TIFF, GIF, JPEG, etc.). An add-on created for annotating TIFF images, for instance, could be a specialization of a more general image annotation add-on, and define additional elements such as *pagenumber*.¹⁰ An add-on created for supporting structured annotations, i.e., annotations allowing users to choose the content of their annotation from a given vocabulary, could restrict the range of a certain model element to a certain vocabulary.

Technically, an add-on is a lightweight software component which can be included into LEMO without modifying already existing code. An add-on must obey a certain contract

⁸ Representational State Transfer (REST): <http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>.

⁹ Annotea Annotation Schema: <http://www.w3.org/2000/10/annotation-ns>.

¹⁰ The TIFF image file format supports multiple pages, in contrast to other image formats such as JPEG.

Fig. 2 The LEMO core schema and its relationship to Annotea

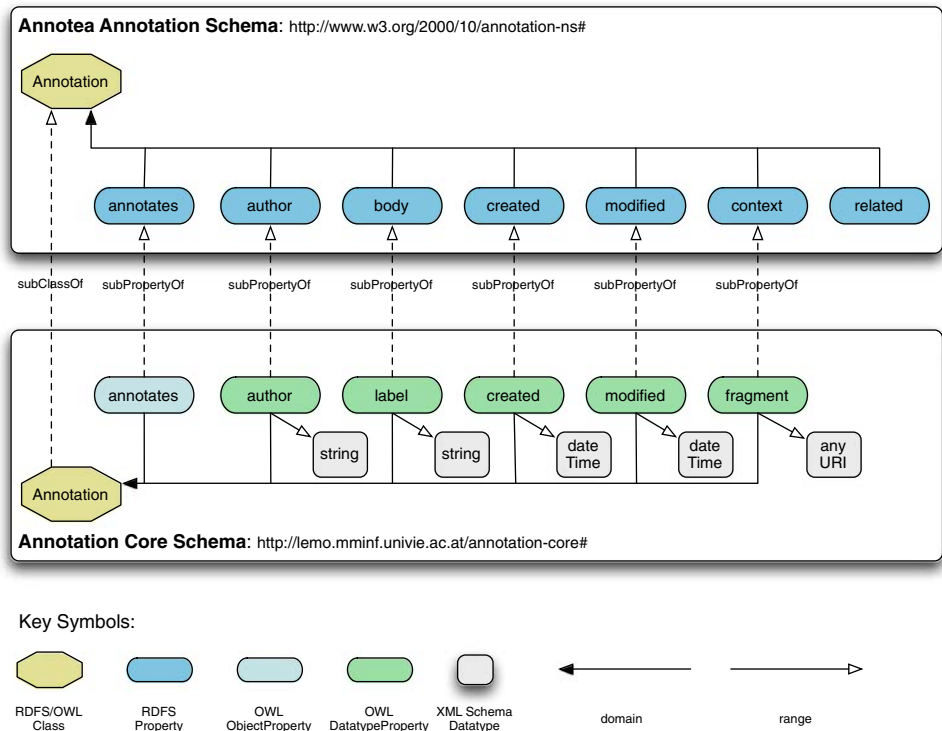
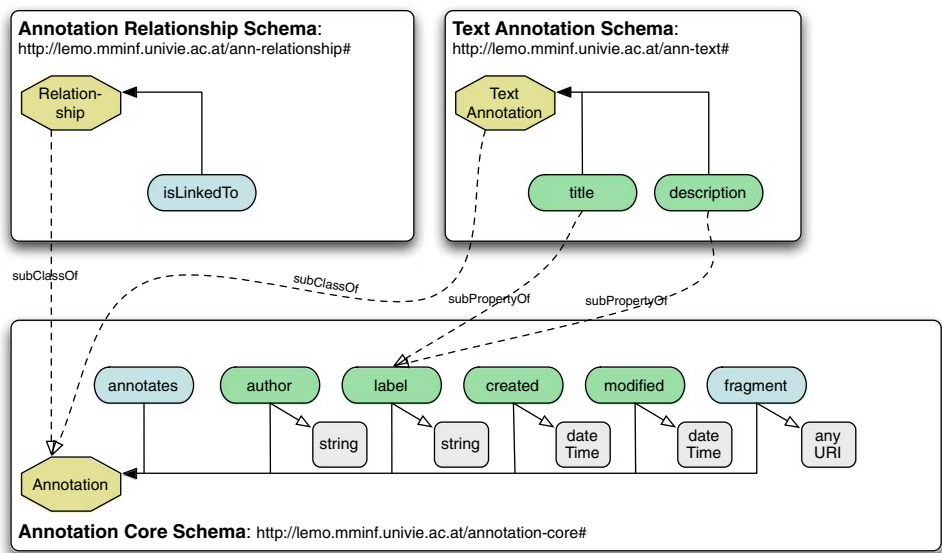


Fig. 3 Extensions to the LEMO core schema to support relationships between annotations (e.g., for discussion threads) and text annotations



which is defined in term of a predefined interface. LEMO currently supports two types of add-ons: content-type and annotation-type add-ons. In Fig. 4, we illustrate the usage of *add-ons* and *annotation profiles*: at the LEMO system core we maintain the so-called *annotation core model*, which defines a set of common elements (e.g., label, author information) required by any kind of annotation type. Extensions to the core model can be defined in terms of annotation profiles which can then be introduced into the LEMO framework by providing and integrating an appropriate content- or application-type specific add-on.

The design of the LEMO annotation model raises the question why we did not simply reuse the Annotea Annotation Schema as it has been defined. First of all, the LEMO approach is not only a conceptual model but also has a technical basis; it is accessed by the indexing mechanism¹¹ and also by the query engine. Therefore, it was necessary to apply restrictions on the ranges (datatype, type) of the annotation properties. The second reason is the notion of context

¹¹ For an index, for instance, the data type of the content value is essential.

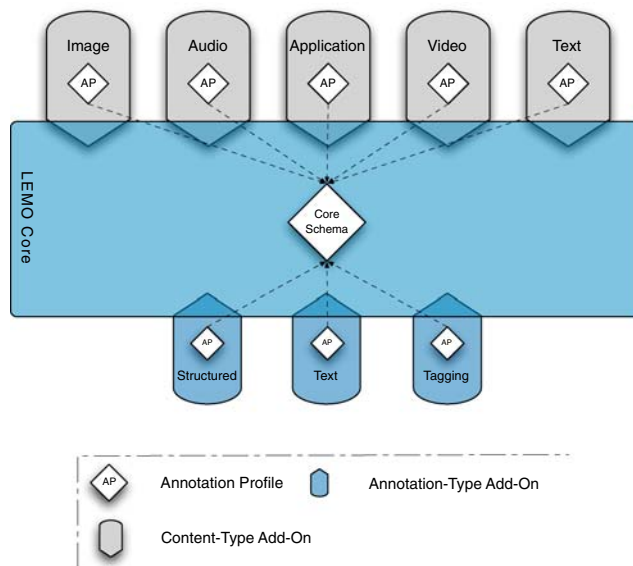


Fig. 4 Add-ons and annotation profiles

in the Annotea Annotation Schema, which we have adapted in order to be interoperable with existing fragment identification standards. In the following section we will further elaborate on this issue.

4.3 Fragment identification

As discussed in Sect. 3.2, it must be possible to address specific parts of media resources in a unified way that can easily be separated from the annotation model. Thereby the system becomes flexible in content-type support and annotation models can be reused. On the Web, the URI fragment identifier is the common and standardized method to refer to a fragment of a resource. Also LEMO uses fragments identified by a URI as a common denominator to facilitate interoperability of fragment definitions without the need to extend the core model.

In the original Annotea Annotation Schema the cardinality of the corresponding `annotates` and `context` properties are not limited; the `context` property has an unspecified range, which can lead to context definitions that have no formal connection to their resource. Schroeter et al. [54] extend the Annotea Annotation Schema to ensure this formal connection between multiple `context` and `annotates` properties. They argue that the schema must be extended because URI-based fragment identification is not suitable for certain content-types. Since their paper has been published, the interest for supporting various media formats has grown. Although the need for direct references to media fragments is a known issue [57], support for various content-types is still limited by now [30].

Recently, the W3C launched a working group to standardize temporal and spatial media fragments on the Web.¹² Apart from the need for a widespread standard for media fragments on the Web, the MPEG community has specified a URI based fragment identification standard. Fragment Identification of MPEG Resources—MPEG-21 FID—is defined in Part 17 of the MPEG-21 framework [28]. It supports all MPEG resources and can be used to address parts of MPEG resources. It is based on the XPointer Framework and adds temporal, spatial and spatio-temporal axis, logical units, byte ranges, masks for videos and items and tracks of ISO Base Media Files [27,28].

We believe that the ongoing efforts of the W3C to provide fragment specifications for media objects on the Web, the released ISO standard *Multimedia framework (MPEG-21)-Part 17: Fragment Identification of MPEG Resources*, together with other projects [49,59] that aim at promoting standards for URI-based fragment identification for temporal media and plain text, create a wide spectrum of content-types that can already be addressed in uniform and standardized ways.

As illustrated in Fig. 2, we choose to limit the range of the `fragment` element in the LEMO core schema to URI, but not to limit the cardinality of objects or fragments that are annotated. If more than one digital item is annotated, the formal connection between the fragment property and its related digital item is the URI excluding the fragment identifier. Listing 1 illustrates part of an annotation on a digital item, which is available at <http://www.univie.ac.at/test.mpg>. It addresses a fragment identified by the URI [http://www.univie.ac.at/test.mpg#mp\(~time\('npt','30','40'\)\)](http://www.univie.ac.at/test.mpg#mp(~time('npt','30','40'))). The *media pointer scheme* (*mp*) of MPEG-21 is used to identify a range given in *normal playtime* (*npt*) starting at 30 s and ending at 40 s of the movie `test.mpg`.

Using the fragment identifier of URIs to address a specific portion of a digital item has pros and cons when compared to having a model that is part of the annotation model. The fragment part of a URI is basically an encoded string. Depending on the fragment scheme, which is specified with a content-type's MIME type registration, handling of information that is encoded into a string is cumbersome, hence unsuitable as internal representation. A dual approach that builds on URI fragment identification and an optional alternative representation can be realized by extending the annotation core schema. An extension can use its own internal representation while preserving the benefit of interoperable fragment identification via URIs.

Listing 1 shows an MPEG-21 fragment identifier that links to a time segment that starts after 30 s and ends after 40 s of the video resource `test.mpg`. Listing 2 refers to the same

¹² Media Fragments Working Group, <http://www.w3.org/2008/WebVideo/Fragments/>.

```

<rdf:RDF
  xmlns:a="http://lemo.mminf.univie.ac.at
/annotation-core#"
  ...">
  ...

  <a:annotates>http://www.univie.ac.at/
test.mpg</a:annotates>
  <a:fragment>
    http://www.univie.ac.at/test.mpg#mp
    (~time('npt','30','40'))
  </a:fragment>
  ...
</rdf:RDF>

```

Listing 1 Time fragment of a video expressed according to the MPEG-21 fragment identification specification

```

<rdf:RDF
  xmlns:a="http://lemo.mminf.univie.ac.at
/annotation-core#"
  xmlns:x="http://lemo.mminf.univie.ac.at
/annotation-video#"
  ...">
  ...

  <a:annotates>http://www.univie.ac.at/
test.mpg</a:annotates>
  <a:fragment>
    http://www.univie.ac.at/test.mpg#mp
    (~time('npt','30','40'))
  </a:fragment>
  <x:time_fragment xmlns:mpeg21="http://
lemo.mminf.univie.ac.at/
annotation_mpeg21#">
    <mpeg21:uri_fid>http://www.univie.
ac.at/test.mpg#mp(~time('npt
','30','40'))
    </mpeg21:uri_fid>
    <mpeg21:time_scheme>npt</mpeg21:
time_scheme>
    <mpeg21:start_time>30</mpeg21:
start_time>
    <mpeg21:end_time>40</mpeg21:
end_time>
  </x:time_fragment>
  ...
</rdf:RDF>

```

Listing 2 Alternative representation within the add-on model

fragment, but adds an expanded representation of the fragment to the extended model using a different namespace declaration (<http://lemo.mminf.univie.ac.at/annotation-video#>). In addition to providing better readability, it facilitates the query process by allowing one to use the already existing SPARQL query interface. Apart from the mandatory `fragment` element, it is up to the extension to determine how to handle dual representations.

Fragment definitions are only useful if a user application can interpret their meaning. This limitation holds for

all approaches, but URI fragments have a standardized and widespread fallback behavior that is by default useful to retain a minimum relationship. If a fragment identifier cannot be processed by a user application, the fragment part of the respective URI is ignored and the requested resource is returned. With the limitation of losing the exact fragment, this behavior preserves the relationship to the resource as a whole.

By using this simple method in LEMO, we aim at improving the interoperability of fragment identification representations in diverse annotation systems. We believe, as Geurts et al. [21] have concluded, that the ubiquitous use of URIs will help to solve the problem of defining interoperable, explicit links between resources and their annotations.

4.4 Exposing annotations as web resources

Since we follow a REST-based approach, the annotation URIs (e.g., <http://www.example.org/annotations/1>) are in fact dereferencable URIs, which can be looked up by humans and machines. Therefore the LEMO Annotation framework fulfills the first and second linked data principles, as described in Sect. 3.3.

To fulfill the third principle, LEMO must be able to expose annotation data in different formats than RDF. Humans typically access Web resources using a browser, which in turn requires an (X)HTML representation in order to display the returned information. We fulfill that requirement by relying on content negotiation, which is a built-in HTTP feature. Figure 5 illustrates how annotations can be retrieved in various formats by specifying the appropriate mime-type in the HTTP `Content-Type` header field. LEMO forwards client requests for a specific annotation (e.g., <http://example.com/annotations/1>) to the appropriate physical representation, i.e., <http://example.com/annotations/html/1> for HTML requests and <http://example.com/annotations/rdf/1> by sending an HTTP 303 See Other response back to the client.

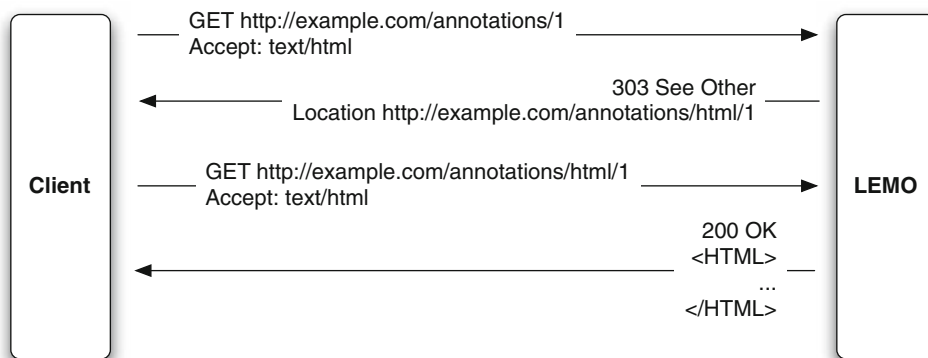
The fourth linked-data principle is fulfilled by the inherent nature of annotations: as already mentioned in Sect. 3.1, an annotation must contain at least the link to the digital item it annotates. This could be, for instance, any multimedia digital item that is exposed on the Web by a digital library system and therefore referencable via its URI or an already existing annotation exposed by the LEMO Annotation Framework.

5 Existing annotation tool implementations

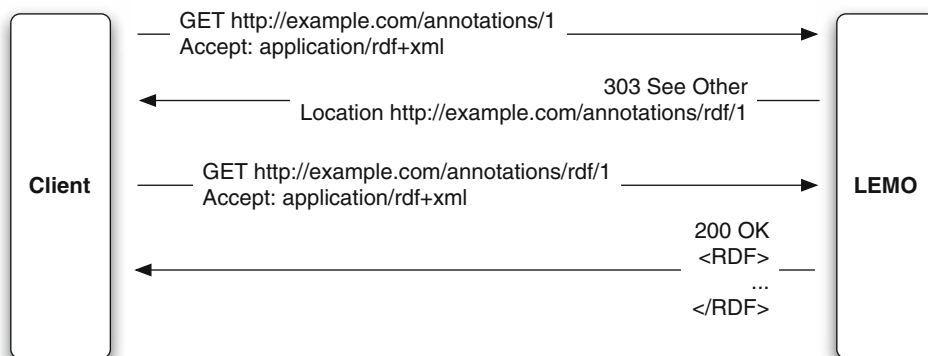
The LEMO Annotation Framework takes the role of a middleware that can be integrated with various storage backends and serve as controller component for various types of front-end annotation user interfaces. In this section, we first focus on the architectural details of LEMO. Thereafter,

Fig. 5 Retrieving annotations in RDF and HTML, respectively

Dereference an annotation URI, requesting HTML



Dereference an annotation URI, requesting RDF content



we briefly describe three different annotation tools that have been implemented on-top of LEMO.

5.1 Implementation: annotation middleware

The first LEMO prototype is implemented in Java, and fulfills the role of the controller in the MVC model. All annotation frontends (viewers in the MVC model) use the annotation middleware to create, update, delete and search annotations. The purpose of the annotation middleware is to keep the frontends independent of any particular back-end implementation. The use of a standardized protocol and exchange format between the middleware and the annotation frontends further increases the reusability of the frontends (see Annotea¹³).

Thus, the annotation middleware provides flexibility in terms of the annotation model and also reduces the development effort for the frontends since the access to the particular back-end has to be implemented only once as part of the middleware. In other words, this approach ensures the extensibility aspect demanded by the LEMO framework.

Requests can be used by all frontends and can also be issued from an internet portal to the annotation middleware directly (again through a proxy server). The output format

can be customized, as the middleware is able to transform formats, in case Annotea is not the desired format for the portal in question. A simple HTTP-based content negotiation, as also suggested by the REST approach, is supported for GET methods, providing LEMO-required linkable annotation resources. In the currently deployed prototype the output is a simple XML containing, among other attributes, the annotation title and item URL.

Figure 6 gives an overview of the LEMO architecture and illustrates its role as annotation middleware. It shows that it can be integrated with various annotation storages (e.g., Fedora, Sesame) and that it supports various Web-based annotation tools (e.g., Image, Video, HTML Annotations). All annotations managed by the LEMO Annotation Framework are also exposed on the Web and can therefore be accessed by ordinary HTML browsers or any other application that supports HTTP and RDF. Additionally, annotations can be queried using the SPARQL query language.

5.1.1 Annotation storage

The annotation persistence layer is defined by an abstract interface class, which allows flexibility and extensibility in the back-end implementation.

¹³ <http://www.w3.org/2001/Annotea/User/Protocol.html>.

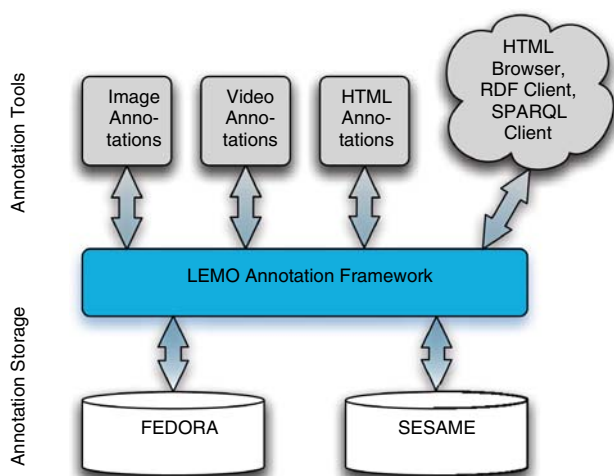


Fig. 6 LEMO architecture overview

We have developed two implementations of the annotation persistence interface; the first is built on top of the Sesame RDF middleware.¹⁴ This has the advantage of providing direct support for RDF query languages (e.g., SPARQL) which in turn allows our annotation repository to serve as a Semantic Web data source as part of the LEMO approach to open and interoperable systems. A second implementation is built on top of the open source Fedora¹⁵ repository.

5.1.2 Authentication

Because the frontends for the annotation services are separated from the database where the annotations are stored, the access to the database is a separate service. In that case storing the data requires authentication of the user. Authentication is done by the service provider that offers the database to be annotated. For TEL the annotation service is accessed via a TEL proxy service that checks for user authentication.

The annotation service does IP authorization and only allows requests from the TEL proxy server. This proxy service will provide the user-parameter when invoking the service.

5.2 Annotation tools

The underlying middleware supports the LEMO requirements of a uniform annotation model and uniform fragment identification that are necessary for multimedia content support. It is however clear that different media types also require different user interfaces for handling the media-specific aspects of digital items (for example, time-based segmentation). We have implemented two media-specific user

interfaces for images and video content and, through our implementation of the Annotea standard, support existing HTML annotation tools as well.

5.2.1 Image annotations

In the context of our image annotation tool, an image is any web resource (i.e., identified by a URL) that can be displayed as an image in a web browser, which is in fact browser-dependent.

The image annotation interface is a browser-independent Java + Javascript application that was developed using the Google Web Toolkit.¹⁶ The interface supports zooming and panning of images, a variety of fragment definitions (point, ellipse, rectangle, polygon, and freehand), and annotation threading. A screenshot of the prototype image annotation tool is shown in Fig. 7.

In the case of images, the fragment URI is defined using the MPEG-21 approach to spatial addressing, as discussed in Sect. 3.2. The media-specific fragment extension of the image annotation class is described in the SVG.¹⁷ An SVG definition of the image fragment, serialized as XML, is embedded in the Annotea RDF tag. The advantage of the SVG extension is that image fragments can be directly viewed in the latest browsers when an annotation is accessed as a linked web resource, with no additional software interpretation required.

For image annotations it is technically possible to store the bitstream of the image together with the annotation or to store only the URL of the image together with the annotation. The first case requires more storage but the advantage is that one is not relying on the persistency of other external images and videos. However, this case raises copyright violation issues; hence the default setting of the image annotation middleware is to save references only, as with the other media types.

5.2.2 Video annotations

Browser support for displaying video content has crystallized as Flash plugin technology, due largely to the predominance of YouTube as a video hosting service. This has at the same time made the Flash Video Format, a variant of the H.263 recommendation,¹⁸ a de-facto standard for web video resources. Latest versions of the Flash Video Player also support the MPEG-4 format, more specifically the H.264 standard (ISO/IEC 14496-10).

¹⁶ Google Web Toolkit: <http://code.google.com/webtoolkit/>.

¹⁷ Scalable Vector Graphics (SVG): <http://www.w3.org/Graphics/SVG/>.

¹⁸ H.263: Video coding for low bit rate communication: <http://www.itu.int/rec/T-REC-H.263/>.

¹⁴ Sesame RDF framework: <http://www.openrdf.org/>.

¹⁵ Fedora Digital Library System: <http://www.fedora-commons.org/>.

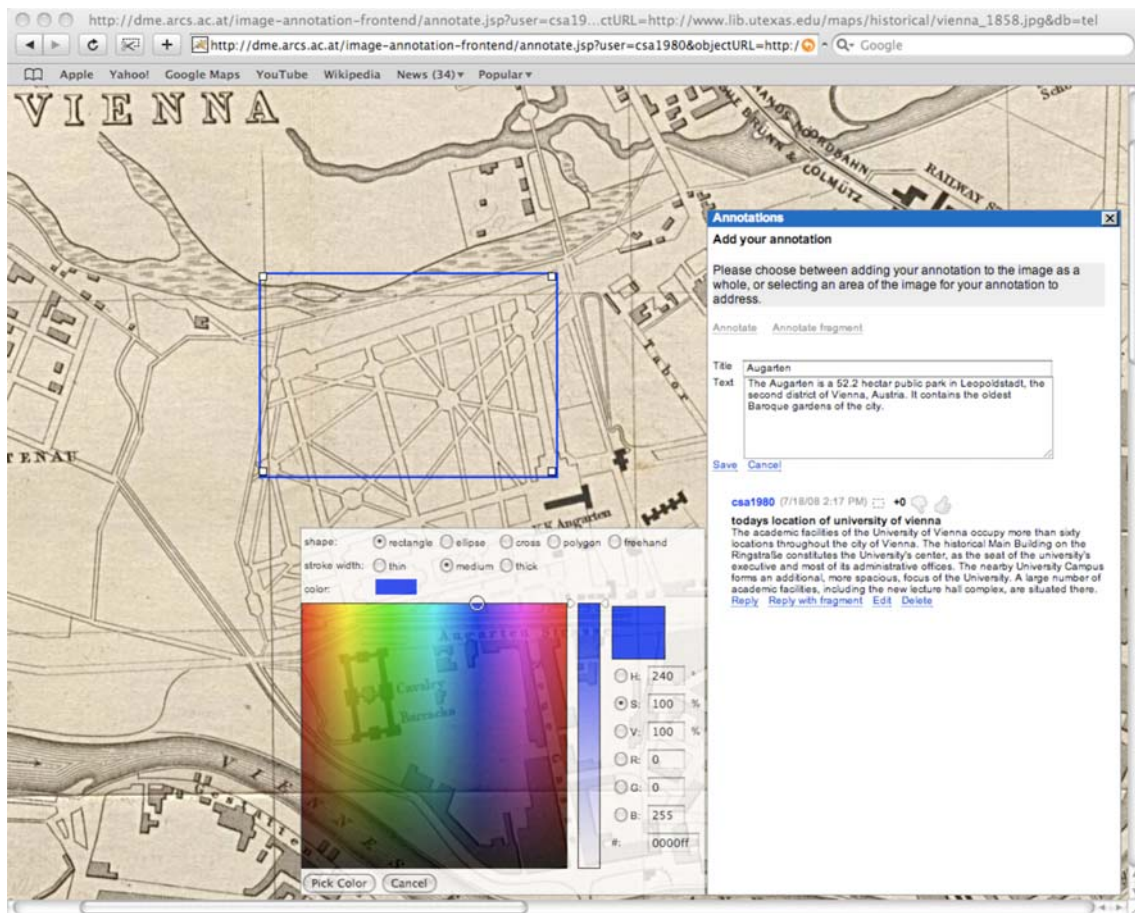


Fig. 7 The TELplus prototype image annotation user interface

This fact compelled us to choose Flash as the technology choice for the user interface. According to the company Adobe, the Flash plugin has achieved a market penetration of 99% in the combined “mature market”, which includes the United States, Europe, and Japan.

The user interface supports the definition of combined spatial fragments and time segments, as well as a video player.

The media-specific fragment extension of the video annotation class is MPEG-21. The MPEG-21 definition of the video fragment, serialized as XML, is embedded in the Annotea RDF description tag.

In principle, we could support a wide variety of video formats by converting existing video files to Flash video and streaming these files from the LEMO server; however, copyright considerations preclude such an approach at this time.

5.2.3 HTML annotations

Because our middleware implements the Annotea protocol, HTML annotations are possible using the Annozilla (<http://annozilla.mozdev.org/>) plugin for the Firefox Web Browser.

Unfortunately, development of this tool has stalled since early 2007 and the plugin does not run under the latest Firefox V3 release.

In any case, it is clearly desirable to have a browser-independent approach for HTML annotations; we intend to carry out this work in the context of the TELplus¹⁹ project.

6 Initial evaluation

In this section, we describe our experiences with the LEMO Annotation Framework in a real-world environment and how the design decisions described in this paper can facilitate the development of annotation tools.

6.1 Proof of concept

In order to make a real-world evaluation of our implementation and approach, the LEMO annotation framework will

¹⁹ TELplus project: <http://www.theeuropeanlibrary.org/portal/organisation/cooperation/telplus/>.

be coupled with a test version of the TEL portal. In the TEL scenario, purpose of an annotation service is to enable users to contribute information to digital objects stored in libraries, archives, and museums across Europe. The TEL portal assumes that these digital objects can be hosted anywhere but that they are identified by a unique URL.

As part of the TELplus project, the three annotation front-end services discussed in the previous section, for image, video and HTML annotations will be provided. Each annotation service will have its own user interface which is invoked from within the portal (by a secondary user) through a proxy server (the primary user of our annotation service).

In addition, we assume that the primary consumer of the annotation service is a web portal that aggregates a large number of digital objects, either by hosting them directly or by referencing them. This portal most likely offers a number of value-added services related to the media in question, such as community-building, customization, and sharing. Other well-known examples of such portals are YouTube and Flickr. Furthermore, our service assumes that secondary users (the subscribers to the TEL portal) are identified and authenticated by the portal, and that anonymous annotations are not allowed by TEL (or any other authorized portals). Unique IDs in our implementation are concatenations of the portal IDs, which are unique in our system, together with the delivered user IDs, which are assumed to be unique in the TEL user management system.

As one can see, our approach allows us to offer an annotation service that is only loosely coupled to the portal in question (in contrast to YouTube and Flickr, in which the annotation functionality is an integral part of the portal implementation itself). For portals, this has the primary advantage of enabling the integration of annotation services with minimal implementation effort. Annotations will be stored in our independent database and not as part of the portal metadata.

From the point of view of our annotation service, this means that a single annotation instance can serve multiple portals, which has synergistic effects like faster build-up of the user community and establishment of a critical mass of users. We further hypothesize that the fact that users from various portals can use this annotation service will increase the impact of the annotation database and facilitate navigation between different portals.

6.2 Qualitative evaluation

Until this point in time, we have demonstrated the advantages of the LEMO framework for annotation interface developers in a number of ways.

First, the extensibility of the model has been shown by the simple integration of specific fragment types for different media (images and video). The flexibility of the middleware has been demonstrated by the ease of changing the underlying

persistence layer; implementations based on RDF (Sesame) and Fedora have been provided in a matter of person-days.

Given the easy-to-implement REST-like interface, we have experienced rapid development times for annotation user interfaces, a matter of person-weeks. Naturally, the purely user interface-driven considerations can consume orders of magnitude more resources—but such considerations are independent of any underlying annotation framework. At the same time, by implementing a de-facto standard interface (Annotea), we extend the range of supported annotation frontends immediately to include projects like Annotilla HTML annotation support.

6.3 Open issues

We are aware of the fact that the influence of an annotation framework, which usually represents the back-end of an annotation solution, is minor with respect to user acceptance in electronic environments. At the end, the user interfaces for annotating multimedia contents will be the critical components for the success or failure of an annotation solution. A well-designed user interface can motivate end users to add their annotations and consequently use an annotation framework such as LEMO. This, however, is out of the scope of the work reported in this paper. Here we mainly concentrated on a framework that enables developer and designers to build their annotation solutions in a more interoperable manner as most proprietary back-end solutions would be.

In the coming year, through the described test deployment at TEL, we will carry out an evaluation of our annotation framework from the standpoint of end users. Test users of the TEL portal will deliver important feedback, both directly through questionnaires and indirectly through activity logs. Questionnaires will focus on usability issues, whereas a quantitative analysis of log files, focusing on the questions of user and community behavior and the added-value of annotations will be published. We will also analyze statistics related to the rate of community build-up and cross-linking with external portals, in order to test the hypotheses outlined in the previous section.

7 Conclusions

In this paper, we have presented the LEMO Annotation Framework, which fulfills three main requirements that go beyond those of existing, well-known annotation systems: first, it provides a uniform annotation model for multimedia contents and various types of annotations, second, it can address fragments of various contents type in a uniform, interoperable manner, and third, it pulls annotations out of closed data silos and makes them available as interoperable, dereferencable Web resources. So far, two real-

world annotation tools, among them one for the TEL project (TEL), have been implemented on-top of the LEMO Annotation Framework. We expect further implementations to be realized in the near future.

While there exist other systems that support annotations for multimedia digital items, to the best of our knowledge, there is no other system that integrates the previously mentioned three features into a single solution. The annotation model we have proposed may seem straight-forward because it reuses the Annotea schema to a large extent. We believe, however, that the reuse of elements defined in existing standards is a necessary step towards interoperability. This is also the case for the identification of fragments: while annotation systems can internally treat fragments using their own representation, they should at least follow a standardized format for exchanging those fragment identifiers. Only in that way, external applications can process and interpret annotations that address a certain fragment and not the whole part of a digital item; a feature which we believe is extremely relevant for annotations in general. Last but not least, the Web provides an optimal environment for collaborative tasks such as annotating digital items. By pulling out annotation data from closed data silos and publishing them as reusable, structured data on the Web, we give external applications the opportunity to reuse the annotation information generated instances of the LEMO Annotation Framework.

In our future work, we will focus on extending the LEMO Annotation Framework for additional content- and annotation types. We would like to implement annotation tools for various types of digital items (e.g., PDF documents, online audio files, online slideshows, etc.). Since we believe that fragment identification is an often neglected issue, we aim at contributing to the standardization process for fragment identifiers. Finally, we would like to integrate the idea of annotations on the Web with other initiatives that aim at publishing data on the Web.

Acknowledgments The work reported in this paper was partly funded by the Austrian Federal Ministry of Economics and Labour and the European Union as part of the 6th Framework Program (BRICKS) and the eContentplus program (TELplus). We would like to thank our colleagues Niko Popitsch, Bernhard Schandl, and Stefan Zander for their valuable comments on this paper.

References

1. Adobe Inc.: Adobe acrobat professional. <http://www.adobe.com/products/acrobatpro/>
2. Agosti, M., Benfante, L., Orio, N.: Ipsa: A digital archive of herbals to support scientific research. In: Proceedings of the International Conference on Asian Digital Libraries (ICADL), pp. 253–264 (2003)
3. Agosti, M., Albrechtsen, H., Ferro, N., Frommholz, I., Hansen, P., Orio, N., Panizzi, E., Pejtersen, A.M., Thiel, U.: DiLAS: a digital library annotation service. In: Proceedings of Annotation for Collaboration—A Workshop on Annotation Models, Tools and Practices (2005)
4. Agosti, M., Ferro, N.: Annotations: Enriching a digital library. In: Koch, T., Sølvberg, I. (eds.) ECDL, Lecture Notes in Computer Science, vol. 2769, pp. 88–100. Springer, Berlin (2003)
5. Agosti, M., Ferro, N.: Annotations as context for searching documents. In: Crestani, F., Ruthven, I. (eds.) CoLIS, Lecture Notes in Computer Science, vol. 3507, pp. 155–170. Springer, Berlin (2005)
6. Agosti, M., Ferro, N.: Search strategies for finding annotations and annotated documents: The fast service. In: Larsen, H.L., Pasi, G., Arroyo, D.O., Andreasen, T., Christiansen, H. (eds.) FQAS, Lecture Notes in Computer Science, vol. 4027, pp. 270–281. Springer, Berlin (2006)
7. Agosti, M., Ferro, N.: A formal model of annotations of digital content. *ACM Trans. Inf. Syst.* **26**(1) (2007)
8. Agosti, M., Ferro, N., Frommholz, I., Thiel, U.: Annotations in digital libraries and laboratories—facets, models and usage. In: Heery, R., Lyon, L. (eds.) ECDL, Lecture Notes in Computer Science, vol. 3232, pp. 244–255. Springer, Berlin (2004)
9. Baker, T., Dekkers, M., Heery, R., Patel, M., Salokhe, G.: What terms does your metadata use? Application profiles as machine-understandable narratives. *J. Digit. Inform.* **2**(2) (2001). <http://jodi.ecs.soton.ac.uk/Articles/v02/i02/Baker/>
10. Berners-Lee, T.: Linked data (2006). <http://www.w3.org/DesignIssues/LinkedData.html>
11. Bottoni, P., Leviaidi, S., Labella, A., Panizzi, E., Trinchese, R., Gigli, L.: Madcow: a visual interface for annotating web pages. In: AVI '06: Proceedings of the Working Conference on Advanced Visual Interfaces, pp. 314–317. ACM Press, New York (2006)
12. Brocks, H., Stein, A., Thiel, U., Frommholz, I., Dirsch-Weigand, A.: How to incorporate collaborative discourse in cultural digital libraries. In: Proceedings of the ECAI 2002 Workshop on Semantic Authoring, Annotation & Knowledge Markup (SA-AKM02), Lyon, France (2002)
13. Brush, A.J.B., Barger, D., Gupta, A., Cadiz, J.J.: Robust annotation positioning in digital documents. In: CHI, pp. 285–292 (2001)
14. Bush, V.: As we may think. *Atl. Mon.* **176**(1), 101–108 (1945)
15. Candela, L., Castelli, D., Ferro, N., Ioannidis, Y., Koutrika, G., Meghini, C., Pagano, P., Ross, S., Soergel, D., Agosti, M., Dobrev, M., Katifori, V., Schuldt, H.: The DELOS digital library reference model—foundations for digital libraries, version 0.98. Tech. rep., DELOS—A Network of Excellence on Digital Libraries (2007). http://www.delos.info/files/pdf/ReferenceModel/DELOS_DLReferenceModel_0.98.pdf
16. DCMI: Dublin Core Metadata Element Set, Version 1.1. Dublin Core Metadata Initiative (2006). <http://dublincore.org/documents/dces/>
17. Denoue, L., Vignollet, L.: An annotation tool for web browsers and its applications to information retrieval (2000)
18. Frisbe, M.E.: Searching for information in a hypertext medical handbook. In: Hypertext '87 Proceedings, November 13–15, 1987, Chapel Hill, North Carolina, USA, pp. 57–66. ACM, New York (1987)
19. Frommholz, I., Fuhr, N.: Probabilistic, object-oriented logics for annotation-based retrieval in digital libraries. In: JCDL '06: Proceedings of the 6th ACM/IEEE-CS Joint Conference on Digital Libraries, pp. 55–64. ACM, New York, NY, USA (2006)
20. Frommholz, I., Fuhr, N.: Probabilistic, object-oriented logics for annotation-based retrieval in digital libraries. In: Marchionini, G., Nelson, M.L., Marshall, C.C. (eds.) JCDL, pp. 55–64. ACM, New York (2006)
21. Geurts, J., van Ossenbruggen, J., Hardman, L.: Requirements for practical multimedia annotation. In: Workshop on Multimedia and the Semantic Web, pp. 4–11. Heraklion, Crete (2005). <http://www.cwi.nl/~media/publications/MMSemweb2005.pdf>

22. Gonçalves, M.A., Fox, E.A., Watson, L.T., Kipp, N.A.: Streams, structures, spaces, scenarios, societies (5s): a formal model for digital libraries. *ACM Trans. Inf. Syst.* **22**(2), 270–312 (2004)
23. Google Inc.: Google notebook. <http://www.google.com/notebook/>
24. Grahl Software Design: Pdf annotator. <http://www.ograhl.com/en/pdfannotator/>
25. Heery, R., Patel, M.: Application profiles: mixing and matching metadata schemas (2000). <http://www.ariadne.ac.uk/issue25/app-profiles/>
26. Hunter, J., Schroeter, R., Guerin, J., Henderson, M.: Vannotea. <http://www.itee.uq.edu.au/~eresearch/projects/vannotea/index.html>
27. ISO/IEC: Introducing MPEG-21 Part 17—An Overview. International Organization for Standardization—MPEG Working Group (ISO/IEC JTC 1/SC 29/WG 11/N7221) (2005). <http://www.chiariglione.org/mpeg/technologies/mp21-fid/index.htm>
28. ISO/IEC: *Multimedia framework (MPEG-21)—Part 17: Fragment Identification of MPEG Resources*. (International Organization for Standardization, Geneva 2006)
29. Jacobs, I., Walsh, N.: Architecture of the World Wide Web, Volume One (2004). <http://www.w3.org/TR/webarch/>
30. Jochum, W.: Requirements of fragment identification. In: International Conferences on New Media Technology and Semantic Systems, pp. 172–179. Springer, Berlin (2007)
31. Jochum, W., Kaiser, M., Schellner, K., Wirl, F.: Living memory annotation tool—image annotations for digital libraries. In: Kovács, L., Fuhr, N., Meghini, C. (eds.) ECDL, Lecture Notes in Computer Science, vol. 4675, pp. 549–550. Springer, Berlin (2007)
32. Kahan, J., Koivunen, M.R.: Annotea: an open RDF infrastructure for shared Web annotations. In: WWW '01: Proceedings of the 10th International Conference on World Wide Web, pp. 623–632. ACM Press, New York (2001)
33. Kautz, H., Selman, B., Shah, M.: Referral web: combining social networks and collaborative filtering. *Commun. ACM* **40**(3), 63–65 (1997). <http://doi.acm.org/10.1145/245108.245123>
34. Koivunen, M.R.: Semantic authoring by tagging with annotea social bookmarks and topics. In: The 5th International Semantic Web Conference (ISWC2006)—1st Semantic Authoring and Annotation Workshop (SAAW2006), Athens (2006)
35. Kruk, S.R., Decker, S., Decker, L.Z.S., Zieborak, L.: Jeromedl—adding semantic web technologies to digital libraries. In: Andersen, K.V., Debenham, J.K., Wagner, R. (eds.) DEXA, Lecture Notes in Computer Science, vol. 3588, pp. 716–725. Springer, Berlin (2005)
36. Lagoze, C., Payette, S., Shin, E., Payette, C.W.S., Shin, E., Wilper, C.: Fedora: an architecture for complex objects and their relationships. *Int. J. Digit. Libr.* **6**(2), 124–138 (2006)
37. Library of Congress: Library of Congress Subject Headings (LCSH) (2007). <http://www.loc.gov/aba/cataloging/subject/>
38. Marshall, C.C.: Annotation: from paper books to the digital library. In: DL '97: Proceedings of the Second ACM International Conference on Digital Libraries, pp. 131–140. ACM Press, New York (1997)
39. Marshall, C.C.: Toward an ecology of hypertext annotation. In: UK Conference on Hypertext, pp. 40–49 (1998)
40. Marshall, C.C.: The future of annotation in a digital (paper) world. In: Twidale, H. (eds.) Successes and Failures of Digital Libraries, pp. 97–117. Urbana-Champaign, University of Illinois (2000). <http://www.csdl.tamu.edu/~marshall/uiucpaper-complete.pdf>
41. Marshall, C.C., Brush, B.A.J.: Exploring the relationship between personal and public annotations. In: JCDL '04: Proceedings of the 4th ACM/IEEE-CS Joint Conference on Digital Libraries, pp. 349–357. ACM Press, New York (2004)
42. Maryland Information and Network Dynamics Lab Semantic Web Agents Project (MINDSWAP): Photostuff project homepage. <http://www.mindswap.org/2003/PhotoStuff/>
43. Mojiti Inc.: Mojiti. <http://mojiti.com>
44. Nichols, D.M., Pemberton, D., Dalhoumi, S., Larouk, O., Belisle, C., Twidale, M.: Debora: Developing an interface to support collaboration in a digital library. In: Borbinha, J.L., Baker, T. (eds.) ECDL, Lecture Notes in Computer Science, vol. 1923, pp. 239–248. Springer, Berlin (2000)
45. OCLC Online Computer Library Center: Dewey Decimal Classification (DDC) (2007). <http://www.oclc.org/dewey/>
46. Ovsianikov, I.A., Arbib, M.A., McNeill, T.H.: Annotation technology. *Int. J. Hum.-Comput. Stud.* **50**(4), 329–362 (1999)
47. Payette, S., Lagoze, C.: Flexible and extensible digital object and repository architecture (fedora). In: ECDL '98: Proceedings of the Second European Conference on Research and Advanced Technology for Digital Libraries, pp. 41–59. Springer, London (1998)
48. Petridis, K., Anastasopoulos, D., Saathoff, C., Timmermann, N., Kompatsiaris, Y., Staab, S.: M-ontomat-annotizer: Image annotation linking ontologies and multimedia low-level features. In: Gabrys, B., Howlett, R.J., Jain, L.C. (eds.) KES (3), Lecture Notes in Computer Science, vol. 4253, pp. 633–640. Springer, Berlin (2006)
49. Pfeiffer, S., Parker, C., Pang, A.: Specifying time intervals in uri queries and fragments of time-based web resources. Network Working Group, Internet-Draft (2005). http://www.annodex.net/TR/URI_fragments.html
50. Phelps, T., Wilensky, R.: Robust intra-document locations (2000). <http://www9.org/w9cdrom/312/312.html>
51. Phelps, T.A., Wilensky, R.: Multivalent annotations. In: European Conference on Digital Libraries, pp. 287–303 (1997)
52. Salembier, P., Smith, J.: Mpeg-7 multimedia description schemes. In: IEEE Transactions on Circuits and Systems for Video Technology, vol. 11, pp. 748–759 (2001)
53. Schilit, B.N., Golovchinsky, G., Price, M.N.: Beyond paper: Supporting active reading with free form digital ink annotations. In: CHI, pp. 249–256 (1998)
54. Schroeter, R., Hunter, J., Newman, A.: Annotating relationships between multiple mixed-media digital objects by extending annotea. In: Franconi, E., Kifer, M., May, W. (eds.) ESWC, vol. 4519. Springer, Berlin (2007)
55. The Bricks Community: BRICKS—Building Resources for Integrated Cultural Knowledge Services (2007). <http://www.brickscmmunity.org>
56. The FotoNotes Project: Fotonotes. <http://www.fotonotes.net/>
57. Troncy, R., Harman, L., van Ossenbruggen, J., Hausenblas, M.: Identifying spatial and temporal media fragments on the web. In: W3C Video on the Web Workshop (2007). <http://www.w3.org/2007/08/video/papers>
58. Viddler Inc.: Viddler. <http://www.viddler.com>
59. Wilde, E., Baschnagel, M.: Fragment identifiers for plain text files. In: HYPERTEXT '05: Proceedings of the sixteenth ACM conference on Hypertext and hypermedia, pp. 211–213. ACM Press, New York (2005)
60. Witten, I.H., Boddie, S.J., Bainbridge, D., McNab, R.J.: Greenstone: a comprehensive open-source digital library software system. In: DL '00: Proceedings of the fifth ACM conference on Digital libraries, pp. 113–121. ACM Press, New York (2000). doi:10.1145/336597.336650
61. Wu, X., Zhang, L., Yu, Y.: Exploring social annotations for the semantic web. In: WWW '06: Proceedings of the 15th International Conference on World Wide Web, pp. 417–426. ACM Press, New York (2006)
62. Yahoo Inc.: Flickr. <http://www.flickr.com>
63. Zoomify Inc.: Zoomify annotation system. <http://www.zoomify.com/enterprise.htm#productDetails>

Copyright of *International Journal on Digital Libraries* is the property of Springer Science & Business Media B.V. and its content may not be copied or emailed to multiple sites or posted to a listserv without the copyright holder's express written permission. However, users may print, download, or email articles for individual use.