

A Formal Model of Annotations of Digital Content

MARISTELLA AGOSTI and NICOLA FERRO
University of Padua

3

This article is a study of the themes and issues concerning the annotation of digital contents, such as textual documents, images, and multimedia documents in general. These digital contents are automatically managed by different kinds of digital library management systems and more generally by different kinds of information management systems.

Even though this topic has already been partially studied by other researchers, the previous research work on annotations has left many open issues. These issues concern the lack of clarity about what an annotation is, what its features are, and how it is used. These issues are mainly due to the fact that models and systems for annotations have only been developed for specific purposes. As a result, there is only a fragmentary picture of the annotation and its management, and this is tied to specific contexts of use and lacks general validity.

The aim of the article is to provide a unified and integrated picture of the annotation, ranging from defining what an annotation is to providing a formal model. The key ideas of the model are: the distinction between the meaning and the sign of the annotation, which represent the semantics and the materialization of an annotation, respectively; the clear formalization of the temporal dimension involved with annotations; and the introduction of a distributed hypertext between digital contents and annotations. Therefore, the proposed formal model captures both syntactic and semantic aspects of the annotations. Furthermore, it is built on previously existing models and may be seen as an extension of them.

Categories and Subject Descriptors: H.3.7 [Information Storage and Retrieval]: Digital Libraries

General Terms: Algorithms, Design, Theory

Additional Key Words and Phrases: Annotation, digital content, digital library system, foundations, hypertext

ACM Reference Format:

Agosti M. and Ferro N. 2007. A formal model of annotations of digital content. *ACM Trans. Inform. Syst.* 26, 1, Article 3 (November 2007), 57 pages. DOI = 10.1145/1292591.1292594 <http://doi.acm.org/10.1145/1292591.1292594>

This work was partially supported by the DELOS Network of Excellence on Digital Libraries, as part of the Information Society Technologies (IST) Program of the European Commission (Contract G038-507618).

Authors' addresses: Department of Information Engineering, University of Padua, Via Gradenigo 6/a, 35131 Padova (PD), Italy; email: {agosti, ferro}@dei.unipd.it.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or direct commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org. © 2007 ACM 1046-8188/2007/11-ART3 \$5.00 DOI 10.1145/1292591.1292594 <http://doi.acm.org/10.1145/1292591.1292594>

1. INTRODUCTION

Digital Library Management Systems (DLMS) are currently in a state of evolution: today they are simply places where information resources can be stored and made available to end users, whereas tomorrow they will increasingly become an integrated part of the way the user works. For example, instead of simply downloading a paper and then working on a printed version, a user will be able to work directly with the paper by means of the tools provided by the DLMS and share their work with colleagues. By doing this, the user's intellectual work and the information resources provided by the DLMS can be merged to form a single working context. The DLMS, therefore, is no longer perceived as something external to the intellectual production process, nor is it seen as a mere consulting tool; instead it becomes an intrinsic and active part of the intellectual production process, as pointed out in Agosti and Ferro [2005a], and Candela et al. [2006].

This turning point in DLMS also clearly emerges from the outcomes of the third brainstorming meeting, organized by DELOS¹, the European Network of Excellence on Digital Libraries funded by the EU's 6th Framework Programme. The main conclusions were the following: first, digital libraries need to become more user-centered; second, digital libraries should not simply be passive repositories, rather they should provide users with tools for more active cooperation and communication; and third, there is an increasing need for generalized digital library management systems [DELOS 2004].

Annotations are an effective means to enable the interaction between users and the DLMS we envision, since their use is a diffuse and very well-established practice. Annotations are not only a way of explaining and enriching an information resource with personal observations, but also a means of transmitting and sharing ideas to improve collaborative work practices. Furthermore, annotations allow users to naturally merge and link personal contents with the information resources provided by the DLMS so that a common context unifying all of these contents can be created.

Furthermore, annotations cover a very broad spectrum, because they range from explaining and enriching an information resource with personal observations to transmitting and sharing ideas and knowledge on a subject. Therefore, annotations can be geared not only to the individual's way of working and to a given method of study, but also to a way of doing research. Moreover, they may cover different scopes and have different kinds of annotative context: they can be private, shared, or public, according to the type of intellectual work that is being carried out. In addition, the boundaries among these scopes are not fixed, rather they may vary and evolve with time. Finally, annotations call for active involvement, the degree of which varies according to the aim of the annotation: private annotations require the involvement of the authors, whereas shared or public annotations involve the participation of a whole community. Therefore, annotations are suitable for improving collaboration and cooperation among users.

¹<http://www.delos.info/>

As pointed out by Ioannidis et al. [2005], this turning point of *Digital Library (DL)* requires that “DL development must move *from an art to a science* [and it needs] unifying and comprehensive theories and frameworks across the lifecycle of DL information” [Ioannidis et al. 2005, p. 266]. The *Streams, Structures, Spaces, Scenarios, Societies (5s)* model, proposed by Gonçalves et al. [2004a], is an example of such a framework and was one of the first efforts in this direction. More recently, the reference model for DLMS [Agosti et al. 2006a] aims at laying the foundations and identifying the cornerstone concepts within the universe of digital libraries, thus facilitating the integration of research and proposing better ways of developing appropriate systems; the notion of annotation has been explicitly introduced in this model as a first class concept for the universe of DLs.

The aim of this article is to contribute to the development of such unifying frameworks by proposing a formal model for the annotation of digital contents. The motivations of this proposal lie in the previous presentation: DLs are moving towards more mature DLMS, which are supported by well defined formal frameworks, and annotations are also headed in that broad direction. Furthermore, to date there has been little agreement about what an annotation is, nor has a comprehensive and formal model of the annotation been proposed. With respect to this last point, Buneman et al. [2002, p. 150] state that:

view annotation² is becoming an increasingly useful method of communicating meta-data among users of shared scientific data sets, and to our knowledge, there has been no formal study of this problem.

and Bottoni et al. [2003, p. 216] point out that:

strangely enough, there is not an agreement yet on the definition of digital annotation, or on how to distinguish it from other digital entities (e.g. hyperlinks, metadata, newsgroup messages). Furthermore, an analysis of the basic operations, to be enabled by a digital annotation system, seems to be lacking.

The aim of the formal model we propose is to formalize the main concepts concerning annotations and to define the relationships between annotations and annotated information resources. Therefore, the proposed formal model captures both syntactic and semantic aspects of the annotations, as well as building on previously existing models, such as the *Streams, Structures, Spaces, Scenarios, Societies* model. This new model thus becomes as compatible as possible with the previous ones and may be seen as an extension of them.

The rest of this article is organized as follows: Section 2 provides an overview of annotations and their use in different contexts, thus presenting the reader with some background knowledge and the main issues concerning annotations; Section 3 highlights the key points about annotations that have to be taken into consideration when modeling them, and introduces the different

²The term *view annotation* is due to the fact that the paper specifically deals with annotations on relations of a database and annotations are modeled as a particular kind of view, where view has the usual meaning adopted in the database management field.

areas covered by the proposed formal model; Sections 4 to 8 explain and formalize the various concepts of the formal model, according to what has been anticipated in Section 3; Section 9 capitalizes on the definitions introduced in the previous sections and proposes a comprehensive definition of annotation; Section 10 shows how the notion of hypertext between annotated objects and annotations follows from the proposed definition of annotation; finally, Section 11 draws some conclusions and discusses possible directions for future work.

2. BACKGROUND ON ANNOTATIONS

Over the years, a lot of research has been done on annotations. The main focus of this work has been on the employment of ad hoc devices, or handheld devices that enable reading appliances with annotation capabilities [Marshall et al. 1999, 2001a; Marshall and Ruotolo 2002; Schilit et al. 1998], and the design and development of document models and systems that support annotations [Agosti et al. 2007a; Phelps and Wilensky 1996, 1997, 2000a, 2000b, 2001; Bottoni et al. 2003] in specific management systems, in particular:

- in the Web [Handschuh and Staab 2003; Bottoni et al. 2004, 2005a, 2005b; Brush et al. 2001, 2002; Davis and Huttenlocher 1995; Nagao 2003; W3C 2005a, 2005b, 2007],
- in digital libraries [Agosti et al. 2003, 2005a; Agosti and Ferro 2005b; Agosti et al. 2005b; Agosti and Ferro 2003a; Agosti et al. 2004; Constantopoulos et al. 2004 Rigaux and Spyrtos 2004; Gueye et al. 2004; Frommholz et al. 2003, 2004; Neuhold et al. 2004; Thiel et al. 2004], and
- in databases [Stein et al. 2002; Bhagwat et al. 2004; Buneman et al. 2001, 2002, 2004; Tan 2004].

The aim of this section is to introduce the two main approaches that have been adopted for dealing with annotations: we can consider them either *meta-data*, which is discussed in Section 2.1, or *content*, which is presented in Section 2.2. This broad distinction in the viewpoints about annotations is also pointed out by Marshall [1998] when she distinguishes between formal versus informal annotations, where the former are metadata and the latter are “marginalia of the sort that we write to ourselves as we read a journal article,” [Marshall 1998, p. 41], which correspond to the content.

This section does not aim at providing a full and exhaustive survey of all the systems offering annotation capabilities that have been developed so far; for this please refer to Agosti et al. [2007a]; Handschuh and Staab [2003]; and Nagao [2003]. Instead, the goal here is to describe some relevant uses and features of annotations, so that in Section 3, these features can be used to gain some insights which will prove useful when developing a formal model for annotations.

2.1 Annotations as Metadata

Annotations can be considered metadata, that is, additional data which relate to an existing content and clarify the properties and semantics of the

annotated content. With this aim in mind, annotations have to conform to some specifications that define the structure, the semantics, the syntax, and even the values that annotations can assume. The recipients of this kind of annotation are both people and computing devices. On the one hand, metadata can benefit people because, if they are expressed in a human-readable form and their format and fields are known, they can be read by people and used to obtain useful and well-structured information about an existing content. On the other hand, metadata offer computing devices the means for automatically processing the annotated contents. Consider, for example, the case of the *Machine Readable Cataloging (MARC)*³ records: not only they are a useful information source for the users of *Online Public Access Catalogs (OPAC)*, which often make MARC records accessible, but they also act as a standard for the representation and communication of bibliographic and related information in machine-readable form, and hence for the automatic processing of this information by computing devices. Note that the examples presented below have the dual use just described: they can be considered useful for both people and computing devices.

A relevant example of this use of annotations is the MPEG-7 standard, formally named “Multimedia Content Description Interface,” which is a standard for annotating and describing multimedia content data [ISO 2004]. To a certain degree MPEG-7 supports the interpretation of the information meaning, which can be passed onto, or accessed by, a device or a computer code. MPEG-7 is not aimed at any application in particular; rather, the elements that MPEG-7 standardizes can support many broad ranges of applications. In this case, annotating a multimedia document means filling in the various fields provided by the MPEG-7 standard in order to describe the features of the object at hand. Similar uses of annotations can be found in the natural language processing field; for example, part of speech tagging consists of annotating each word in a sentence with a tag that describes its appropriate part of speech so as to decide whether a word is a noun, a verb, an adjective, and so on [Jurafsky and Martin 2000; Manning and Schütze 2001]. In both cases, annotations are usually embedded in the annotated digital objects.

A broader example of the use of annotations as metadata is provided by the Semantic Web [W3C 2007] initiative promoted by the *World Wide Web Consortium (W3C)*, which aims at enhancing human-understandable data, namely Web pages, with computer-understandable data, namely metadata, so that “information is given well-defined meaning, better enabling computing devices and people to work in cooperation” [Berners-Lee et al. 2001]. The process of adding metadata to Web pages is called *semantic annotation*, because it involves the decoration of existing data, such as a piece of text whose content is understandable only to people, with semantic metadata that describe that piece of text, so that computers can process it and thus in turn, offer automation, integration, and reuse of data across various applications [Handschuh and Staab 2003]. The Semantic Web makes use of the *Resource Description Framework (RDF)*⁴ as a

³<http://www.loc.gov/marc/>

⁴<http://www.w3.org/RDF/>

syntax for describing and exchanging metadata. In this context, the Annotea project developed by the W3C [Kahan and Koivunen 2001; W3C 2005a] sees annotations as metadata and interprets them as the first step in creating an infrastructure for handling and associating metadata with content, thus leading to the Semantic Web. Annotea predefines annotation types from a list that contains some of the following: comments, notes, explanations, or other types of external remarks that can be attached to any Web document or a selected part of the document, without modifying the document. Annotea uses RDF and *eXtensible Markup Language (XML)*⁵ for describing annotations as metadata, and XPointer⁶ for locating the annotations in the annotated document. Annotea employs a client-server architecture, where annotations reside in dedicated servers, and a specialized browser is capable of retrieving them upon request when visiting a Web page. Koivunen and Swick [2001] and Koivunen et al. [2003] go one step further and employ annotations as an extension of bookmarks to improve cooperation among users: the additional data provided by annotations are exploited to describe, organize, categorize, share, and search for the bookmarks.

In the SCHOLNET⁷ DLMS, Constantopoulos et al. [2004] use annotations as metadata to support communication and interaction within scholarly communities and employ the SIS-Telos⁸ knowledge representation language to express them. They introduce a semantic annotation model, where annotations are treated as documents themselves, the semantics of which is captured by a controlled vocabulary of annotation types. Furthermore, annotations can be translated into records compliant with a subset of *Dublin Core Metadata Initiative (DCMI)*⁹ specification to improve interoperability. Constantopoulos et al. [2004] developed a service for the SCHOLNET system, which offers storage, retrieval and deletion of annotations.

In a similar context, *Imaginum Patavinae Scientiae Archivum (IPSA)* [Agosti et al. 2003, 2005a, 2006c] supports the annotation and personalization of image digital archives. The final goal is to provide end users with tools for performing scientific research on images taken from *illuminated manuscripts* [Canova 1988]. One of the most important aims of the research on illuminated manuscripts is the unveiling of hidden connections among illustrations belonging to different manuscripts. The use of annotations has been proposed as a useful way of accessing a digital archive and sharing knowledge in a cooperative environment. In IPSA, annotations are links that connect one image to another image related to it because illustrations were copied from images in other manuscripts, or they were merely inspired by previous works, or they were directly inspired by nature. IPSA utilizes annotations as metadata because they are drawn from a link taxonomy, which comprises two broad classes. The first deals with *hierarchical* relationships between two images, where one

⁵<http://www.w3.org/XML/>

⁶<http://www.w3.org/XML/Linking>

⁷<http://www.ercim.org/scholnet/>

⁸<http://www.ics.forth.gr/isl/r-d-activities/sis.html>

⁹<http://dublincore.org/>

image somehow depends on an earlier one. The second concerns *relatedness* relationships between two images, where they both share similar properties even though they were created independently. Both classes contain more specialized annotation/link types. IPSA manages and stores annotations in the same archive where the illuminated manuscripts are stored, even if annotations are separated from the annotated digital objects and they do not modify them.

Finally, Rigaux and Spyrtatos [2004] and Gueye et al. [2004] propose a data model for the composition and metadata management of documents in a distributed setting, such as a DLMS. The model enables the creation of *composite documents*, which are made up of either composite documents, or *atomic documents*, which can be any piece of uniquely identifiable material. A set of annotations is associated to each composite document, where they interpret annotations as terms taken from a controlled vocabulary or taxonomy to which all authors adhere. The model provides algorithms to automatically compute the annotations of composite documents, starting from the annotations of its composing atomic documents, by means of a subsumption relation defined within the taxonomy mentioned previously.

Annotations are also used in the context of *DataBase Management Systems (DBMS)* and, in particular, in the case of *curated databases* and *scientific databases*. SWISS-PROT¹⁰ is a curated protein sequence database, which strives to provide a high level of annotation, such as the description of the function of a protein, its domain structure, and so on. In this case, the annotations are embedded in the database and merged with the annotated content. BIODAS¹¹ provides a distributed annotation system, which is a system based on Web servers for sharing lists of annotations across a certain segment of the genome. In this case, annotations are not mixed together with the content they annotate, they are instead separated from it. Annotations have types, methods, and categories. The annotation type is selected from a list of types that have biological significance; the annotation method is intended to describe how the annotated feature was discovered and may include a reference to a program; the annotation category is a broad functional category that can be used to filter, group, and sort annotations [Stein et al. 2002].

Buneman et al. [2001, 2002] investigate the use of annotations with respect to the *data provenance* problem, sometimes also referred to as *data lineage* or *data pedigree*. Data provenance, which is the description of the origins of a piece of data and the process by which it arrived in a database, is undoubtedly an open and challenging research issue in the field of DBMS, As Abiteboul et al. [2005] point out. Buneman et al. [2001] distinguish between *why-provenance*, which explains why a given piece of data is in the database, and *where-provenance*, which explains where a given piece of data comes from. The distinguishing feature of scientific databases is that data needs to be tracked, for example, to know which instruments were used to gather the data, and what their settings were. Moreover, scientists and experts continuously correct and annotate

¹⁰<http://www.expasy.org/sprot/>

¹¹<http://biodas.org/>

the original source data and this, too, is a way of carrying out their research work. Bhagwat et al. [2004] carry on the research into provenance and propose an extension to a relational DBMS and to *Structured Query Language (SQL)* called propagate SQL, which provides a clause for propagating annotations to tuples through queries. They see annotations as information about data such as provenance, comments, or other types of metadata; they envisage the following applications of annotations in DBMS: tracing the provenance and flow of data, reporting errors or remarks about a piece of data, and describing the quality or the security level of a piece of data.

As a final example of the use of annotations as metadata, once more in the context of scientific databases, Buneman et al. [2004] propose an archiving technique for managing and archiving different versions of scientific databases over time. They exploit the hierarchical structure of scientific data to represent the content and the different versions of the database with a tree structure. They attach annotations to the nodes of the tree, annotations that contain time-stamp and key information about the underlying data structure. Therefore, these annotations are metadata about the annotations contained in the database itself. In a sense, we could say that these annotations are meta-metadata; as we can see, they differ from the annotations contained in the database, in that they are metadata about the modifications to the contents of the database over time, while the latter are metadata about genome sequences. On the whole, this annotated tree structure provides an additional data layer that enables the development of efficient algorithms for archiving and searching for the different versions of the database.

2.2 Annotations as Content

Annotations are regarded as additional content that relates to an existing content, meaning that they increase the existing content by providing an additional layer of elucidation and explanation. However, this elucidation does not happen, as in the case of annotations as metadata, by means of some kind of constrained or formal description of the semantics of the annotated object. On the contrary, the explanation itself takes the shape of an additional content that can help people understand the annotated content. However, the semantics of the additional content may be no more explicit for a computing device than the semantics of the annotated content.

This view of annotations is comparable to the activity of reading a document and adding notes to it: explanation and clarification of words or passages of the document by expounding on it, providing a commentary on it, and finally completing it with personal observations and ideas.

Therefore, the final recipients of this kind of annotation are people; because a content annotation does not make the annotated object more readily processable by a computer than the same object without annotations. In fact, from the point of view of a computer, the semantics of content annotations needs to be in some way processed, for example, indexed, before it can be used to deal with the semantics of the annotated object; this is quite different from the case of metadata annotations, which are pieces of information ready to be used for

interpreting the semantics of the annotated object. In contrast, the additional semantics provided by content annotations can offer people useful interpretations and comments for the annotated object, making it easier to understand its hidden facets.

This view of annotations entails an intrinsic dualism between annotation as *content enrichment* and annotation as *stand-alone document*: the former considers annotations as mere additional content regarding an existing document and, as a result, they are not autonomous entities, but in fact rely on previously existing information resources to justify their existence; the latter regards annotations as real documents and autonomous entities that maintain some sort of connection with an existing document. This twofold nature of the annotation can be made clearer by considering how we study a document. First of all, we can annotate some passages that require a further looking into; we can consider this as a sort of “annotation as content enrichment.” Second, we can reconsider and collect our annotations and we can use them as a starting point for a new document; this is an example of “annotation as a stand-alone document.” In this case, the annotation process can be seen as an informal, unstructured elaboration that could lead to a rethinking of the annotated document and to the creation of a new one. Note that both kinds of content annotation are valuable and the boundaries between them may fade into one another with the passing of time; as a consequence, both kinds of content annotations need to be considered as first-class digital objects.

NoteCards [Halasz et al. 1987; Halasz 1988] influenced the successive research in the field of hypermedia/hypertext systems. NoteCards is a hypermedia system designed for helping people to work with ideas: authors, researchers, and intellectual work practitioners can analyze information, construct models, formulate topics, and elaborate ideas by using a network of electronic notecards interconnected by typed links. One of the famous “seven issues” mentioned by Halasz [1988] concerns support for collaborative work: he highlighted how annotations are part of the “activities that form the basis of any collaboration effort” [Halasz 1988, p. 848].

Moving forward in the context of the Web, the CoNote [Davis and Huttenlocher 1995] is a cooperative system for supporting communications within groups of users by using shared annotations on a set of documents. CoNote offers plain text or *HyperText Markup Language (HTML)* [W3C 1999] annotations on Web pages and pays particular attention in structuring annotations on the same part of a document as a tree, in order to ease the discussion among the users by supporting replies to previously inserted annotations. CoNote stores annotations separately from the original Web page and displays them on request by using a standard browser. A similar approach is also adopted by the WebAnn system [Brush et al. 2002], which is aimed at supporting interaction between students and instructors by using annotations; with respect to CoNote it offers users the possibility of more finely tuned annotations.

A recent example of this kind of annotation system in the Web is *Multimedia Annotation of Digital Content Over the Web (MADCOW)* [Bottoni et al. 2004, 2005b]. This system enables multimedia annotation on Web pages

and is based on a client-server architecture. Servers are repositories of annotations to which different clients can connect, while the client is a plug-in for a standard Web browser. MADCOW uses *HyperText Transfer Protocol (HTTP)* [Fielding et al. 1999] as the communication protocol between the annotation servers and the browser plugin; moreover, it assumes that pages are written in HTML in order to annotate Web pages. It supports both private and public annotations and allows different pre-established types of annotations, such as explanation, comment, question, solution, summary, and so on, which are defined according to the *Rhetorical Structure Theory (RST)* [Mann and Thompson 1987]. Note that once annotations have been created, they are assigned a *Uniform Resource Locator (URL)* [Berners-Lee 1994a] and are treated as any other HTML document; therefore, annotations can be made on other annotations.

Annotations as content also find a natural application in the context of a DL. A DL is not only the digital version of traditional libraries and archives, but also offers instruments and services that can go beyond the mere presentation of content stored in digital repositories [Lesk 2005; Witten and Bainbridge 2003; Candela et al. 2006]: annotations can be considered one such instrument. Moreover, as introduced in Section 1, both DLs and their management systems, the DLMS, can greatly benefit from annotations for actively involving users and promoting interaction among them. Ioannidis et al. [2005] also agree with this point when they state that DLMS should provide the means for creating annotations, and should support the storage, selective sharing, and configurable presentation of annotations.

Different layers of annotations can coexist for the same document: a private layer of annotations accessible only by authors of the annotations, a collective layer of annotations, shared by a team of people, and finally a public layer of annotations, accessible to all the users of the digital library. In this way, user communities can benefit from different views of the information resources managed by the DL [Marshall 1997; Marshall and Brush 2002, 2004]. A DL can encourage cooperative work practices, enabling the sharing of documents and annotations, also with the aid of special devices, such as XLibris [Schilit et al. 1998]. Finally, as suggested in [Marshall et al. 2001b; Marshall and Ruotolo 2002], searching, reading and annotating the information resources of a DL can be done together with other activities: for example, working with colleagues. This may also occur in a mobile context, where merging content and wireless communications can foster ubiquitous access to DLMS, improving well established cooperative practices of work, and exploiting physical and digital resources. The wireless context and the small form factor of handheld devices challenge our technical horizons for information management and access. Specialized solutions are required to overcome the constraints imposed by such kinds of devices, as reported in Agosti and Ferro [2003b].

An example of this use of annotations in DLMS, is *Collaboratory for Annotation Indexing and Retrieval of Digitized Historical Archive Material (COLLATE)* [Frommholz et al. 2003; Thiel et al. 2004], which supports the collaboration among film scientists and archivists who are annotating historical film documentation dealing with digitized versions of documents about

European films from the 1920s and 1930s. Such documents are censorship documents, newspaper articles, posters, advertisement material, registration cards, and photos. Annotations support user communities in accessing the information resources provided by the DL in a personalized and customized way: they are *dialog acts*, part of a discourse about film documentation, and constitute the document context, intended as the context of the collaborative discourse in which the document is placed. This collaborative discourse is carried out by allowing annotations to annotate other annotations. Note that COLLATE offers different predefined types of dialog acts, such as elaboration, comparison, argumentation, counterargument, and so on; in this respect, COLLATE opts for a solution similar to the one adopted by MADCOW.

Flexible Annotation Service Tool (FAST) FAST [Agosti and Ferro 2003a, 2004, 2005b, 2005a, 2006; Ferro 2004, 2005] is a flexible system designed to support two different things: various architectural paradigms, such as *Peer-To-Peer (P2P)* or *Web Services (WS)* architectures; a wide range of different DLMS. The flexibility of FAST and its independence from any particular DLMS is a key feature for providing users with a uniform means of interaction with annotation functionalities, without the need for changing their annotative practices only because a user works with different DLMS. FAST supports both users and groups of users with different access permission on annotations and offers three different scopes for the annotations: private, shared, and public. Furthermore, annotations in FAST allow users to merge their personal content with the information resources managed by diverse DLMS: annotations can span and cross the boundaries of a single DLMS, annotating digital objects that are part of different DLs, if users so desire. Finally, this use of annotations gives the users the possibility of linking digital objects that otherwise would have remained separated because they were managed by different DLMS. Ioannidis et al. [2005] recently noted this as an advantage for users and a challenge for the next generation DLMS.

FAST also constitutes the underlying infrastructure of the *Digital Library Annotation Service (DiLAS)* project [Agosti et al. 2005c, 2006d, 2006b], which is an ongoing project in the framework of DELOS, the European Network of Excellence on Digital Libraries. The goal of DiLAS is to design and develop a generic annotation service that can be easily used in different DLMS; DiLAS aims at defining a set of application program interfaces to enable both access to this service from different DLMS, and the creation of different annotation clients and user interfaces embedded in various DLMS. The annotation service will be evaluated as a new way of interacting with a DL and cooperating among DL users and stakeholders. With respect to this last issue, the DiLAS project defines the overall aim of a formative evaluation about how the present design of decentralized annotation services complies with the needs of the prospective users. The overall goal of the evaluation is to investigate the extent to which the annotation system complies with the characteristics, activities, tasks, and environments of users, in order to inspire future developments of annotation services and tools for DLs.

Table I. Summary of the Different Viewpoints about Annotations with Respect to Their Use in the Web, DLMS, and DBMS

		Increasing structure of the annotation →	
		Content	Metadata
Increasing structure of the content ↓	Web	CoNote [Davis and Huttenlocher 1995] MADCOW [Bottoni et al. 2003, 2004, 2005b] NoteCards [Halasz et al. 1987; Halasz 1988] WebAnn [Brush et al. 2002]	Annotea [Kahan and Koivunen 2001; W3C 2005a] Semantic Web [W3C 2007]
	DLMS	COLLATE [Frommholz et al. 2003; Thiel et al. 2004] DiLAS [Agosti et al. 2005c, 2006d, 2006b] FAST [Agosti and Ferro 2003a, 2005b, 2005a, 2006]	Composite Documents [Gueye et al. 2004; Rigaux and Spyrtatos 2004] IPSA [Agosti et al. 2003, 2005a, 2006c] SCHOLNET [Constantopoulos et al. 2004]
	DBMS		BIODAS [Stein et al. 2002] Data Provenance [Buneman et al. 2001, 2002; Bhagwat et al. 2004] Scientific Data Archiving [Buneman et al. 2004]

3. MODELING ANNOTATIONS

3.1 Key points

Table I summarizes the discussion introduced in Section 2 and presents systems along two dimensions. One is the degree of structure of the content and the other is the degree of structure of the annotation. The structure of the content can range from loosely structured documents, as in the case of the Web, to highly structured data, as in the case of a database. Similarly, the structure of the annotation can vary from unstructured or loosely structured annotations, as in

the case of content annotations, to very structured annotations, as in the case of metadata annotations. Note that we have put the NoteCards system under the Web box, even though it was developed before the Web, because a hypermedia system is much more closer to the Web than to DBMS or DLMS.

We can point out that across the different systems there is a very wide range of uses of annotations as a powerful tool for uncovering and clarifying the semantics of the annotated objects. The final recipients of annotations can be computing devices or people. The former is mainly the case of metadata annotations which allow annotated objects to be automatically processed, integrated and reused in different applications, even though these metadata annotations can be understandable and useful for people too. The latter is mainly the case of content annotations which elucidate and expound on an annotated object. Note that, also in this latter case, a computing device can become the recipient of such annotations, provided that some further step of processing is performed, for example, indexing. However, in both cases, the semantics of the annotation itself needs to be taken into consideration and modeled. This can happen formally and precisely by agreeing on metadata standards that describe how annotations should to be interpreted and used; alternatively, support can be provided for identifying different predefined annotation types, perhaps with varying levels of detail.

The medium of the annotation can vary a lot: it can range from textual annotations, to image, audio, and video annotations; in a general setting, we may need to deal with multimedia rich annotations, composed of different parts, each with its own medium. All of these different kinds of media have to be considered and properly modeled, in a uniform way where possible.

Both annotations and annotated objects need to be uniquely identified. Moreover, annotations comprise a temporal dimension that is often not explicit, but which limits the creation of the annotation to the existence of another object. This temporal relationship between the annotation and the annotated object does not mean that the annotation cannot be considered a stand-alone intellectual work, but it does impose a temporal ordering between the existence of an annotated object and the annotation annotating it, which cannot be overlooked. In addition, once we have identified both the annotation and the annotated object, we need to link and anchor the annotation to the part of the annotated object in question. This can happen in a way that mainly depends on the medium of the annotated object. On the whole, we need to model how annotations and annotated objects are uniquely identified and linked together, maybe with a varying degree of granularity in the anchoring, paying particular attention to the temporal dimension that regulates the relationships between annotations and annotated objects.

As far as co operation is concerned, almost all of the analyzed systems show that annotations have great potential for supporting and improving interaction among users, and even among computing devices. Therefore, there is a need for modeling and offering different scopes of annotations, for example, private, shared, or public, and managing the access rights of various groups of users.

Finally, a relevant aspect of annotations is that they can take the part of a hypertext [Agosti et al. 2004; Halasz 1988; Marshall 1998] since they enable

the creation of new relationships among existing objects by means of links that connect annotations together with existing objects, as we will see later in more detail. The hypertext viewpoint about annotations is common to different systems, such as Annotea, MADCOW, and NoteCards in the hypermedia/Web context, or DiLAS, FAST, and IPSA in the DLMS context. Halasz [1988] points out that annotations are one of the activities that form the basis of any collaborative effort, and for which hypermedia systems are ideally suited, while Marshall [1998] considers annotations a natural way of creating and growing hypertexts that connect information resources by actively engaging users. In addition, the hypertext between annotations and annotated objects can be exploited not only for providing alternative navigation and browsing capabilities, but also for offering advanced search functionalities, able to retrieve more and better ranked objects in response to a user query and also by exploiting the annotations linked to them [Agosti and Ferro 2005b]. Moreover, DLMS usually offer some basic hypertext and browsing capabilities based on the available structured data, such as authors or references. On the other hand, DLMS do not normally provide users with advanced hypertext functionalities, where the information resources are linked on the basis of the semantics of their content and *Hypertext Information Retrieval (HIR)* functionalities are available, as in Agosti et al. [1991]. Therefore, annotations can turn out to be an effective way of associating this kind of hypertext to a DLMS to enable the active and dynamic use of information resources. In addition, this hypertext can span and cross the boundaries of the single DLMS, if users need to interact with the information resources managed by diverse DLMS [Agosti and Ferro 2004, 2005a]. This latter possibility is quite innovative, because it offers the means for interconnecting various DLMS in a personalized and meaningful way for the end-user, and, as Ioannidis et al. [2005] point out, this is a major challenge for the next generation DLMS. In conclusion, this hypertext has to be explicitly modeled and taken into consideration when dealing with annotations.

3.2 Modeling Approach

The previous discussion clearly demonstrates that annotation is quite a complex concept comprising a number of different aspects. Therefore, when we attempt to model annotations, we have to work through this complexity, first to identify the main macro-areas of this concept, and second to provide clear definitions of the concepts within each macro-area and of the relationships among these concepts.

Figure 1 provides both an overview of the areas covered and the detail of the definitions introduced within each area. The figure clearly shows how these areas correspond to the very basic issues that emerge when we think about annotations: we need to identify annotations and annotated objects in order to link them together, perhaps providing facilities for supporting cooperation, and we have to deal with both the actual contents of an annotation and the semantics expressed by those contents.

Therefore, the objective of the formal model is to delimit the boundaries of each area and to clearly define the concepts contained in each area together with

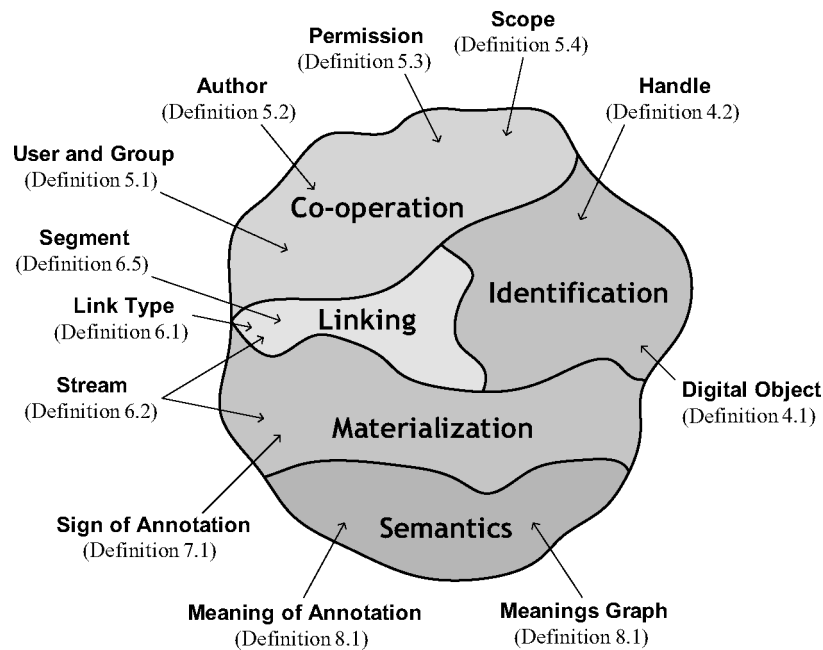


Fig. 1. Overview of the areas covered by the proposed formal model of annotation.

the relationships among them. We can then exploit these definitions to provide a comprehensive definition of annotation and to derive from this definition the notion of hypertext between annotated objects and annotations. To this end, we adopt a modeling approach based on set theory [Halmos 1974] and graph theory [Bollobás 1998; Diestel 2000].

In the remainder of this section, we will briefly introduce the areas shown in Figure 1, which can be used as a map of the concepts dealt with in the formal model. Note that Figure 1 illustrates the main areas from bottom to top in the same order in which they were discussed in Section 3.1, which represents the natural way in which they occur when we reason about annotations. In contrast, the areas are presented below in the order in which they will be discussed in detail in Sections 4 to 8. This order derives from the need to introduce concepts and definitions in a linear fashion, thus avoiding back and forth references between them. Finally, Section 9 will define the annotation itself and Section 10 will discuss the hypertext between annotated objects and annotations, which can also be thought of as a kind of a concise view of the main relationships between annotations and annotated objects.

Identification is the problem of uniquely identifying both the annotation and the annotated objects, highlighting the temporal constraints between them. This area is built around the concept of *handle*, which is defined as a unique identifier of both digital objects and annotations, and the proposed notation for dealing with the time dimension involved by annotations.

Cooperation is about annotations as a cooperation tool among users. It introduces the definitions of *user* and *group* of users, together with the associated concept of *scope* of annotation and *access permission*, which regulate the access policies for a given annotation.

Linking deals with the allowed linking patterns between digital objects and annotations, and the problem of correctly anchoring annotations to digital objects. It defines the concepts of *link type*, which is defined as the allowed methods of linking annotations to annotated objects, *stream*, which abstracts the notion of content of a digital object, and *segment*, which represents a given portion of a stream, useful for anchoring an annotation to a digital object.

Semantics concerns the meaning of the annotation and what it stands for, trying to make explicit the semantics of the different parts of the content of an annotation. It introduces the notions of *meaning of annotation*, which is part of the semantics of the whole annotation, and *meanings graph*, that is a graph allows for interoperability between the different meanings.

Materialization deals with the way in which the semantics carried by an annotation can take shape, that is, the actual content of the annotation perceived by the user. It describes the *sign of annotation*, which is a particular type of stream representing part of the content of an annotation.

The proposed formal model was first introduced in Ferro [2004] and a reduced version of it was adapted to the context of illuminated manuscripts in Agosti et al. [2006c].

4. IDENTIFICATION

In order to uniquely identify both the annotation and the annotated objects, we need to proceed as follows: first, we need to define the objects we deal with, as described in Section 4.1; then, we also have to be able to deal with objects whose relationships are constrained by a temporal dimension, as explained in Section 4.2; finally, a suitable identification mechanism has to be provided, as introduced in Section 4.3.

4.1 Document, Annotation, and Digital Object Sets

According to widely accepted terminology, we adopt the term digital object to refer to information resources managed by an *Information Management System (IMS)*. Indeed, Paskin [2006, p. 6] defines the digital object as “a data structure whose principal components are digital material, or data, plus a unique identifier for this material.” Gonçalves et al. [2004a, p. 292] say that “information in digital libraries is manifest in terms of *digital objects*, which can contain textual or multimedia content (e.g., images, audio, video), and *metadata*,” and they define a digital object as a tuple constituted by a unique handle, structured contents, and structural metadata [Gonçalves et al. 2004a, p. 294]. Finally, Bottoni et al. [2003, p. 217] define the digital object as a typed tuple of attribute-value pairs with at least two mandatory attributes: a unique identifier, and the actual content of the digital object; furthermore, they consider annotations to be

digital objects with specific attributes, that is, annotations are specialized digital objects.

In the following, we need terminology to distinguish between two kinds of digital objects: the generic ones managed by the IMS, which we call *documents*, and the ones that are *annotations*. Therefore, when we use the generic term digital object, we mean a digital object that can be either a document or an annotation. Note that the term “document” is used here in a broad sense, since it indicates a generic multimedia, possibly compound, digital object; examples of such a broad use of the term document can be found in Castelli and Pagano [2002a], who define the document as a structured multilingual and multimedia information object, or in the *Document Object Model (DOM)* [W3C 1998] where the term document indicates “many different kinds of information that may be stored in diverse systems, and much of this would traditionally be seen as data rather than as documents.”

Finally, note that when we talk about annotations we mean both metadata annotations, introduced in Section 2.1, and content annotations, explained in Section 2.2; therefore, we consider both of these types of annotations as kinds of digital objects.

The following definition introduces the different sets of digital objects we will need to deal with.

Definition 4.1. Let us define the following sets:

- D is a **set of documents** and $d \in D$ is a generic **document**. U_D is a **universe set of documents**, which is the set of all the possible documents, so that $D \subseteq U_D$.
- A is a **set of annotations** and $a \in A$ is a generic **annotation**. U_A is a **universe set of annotations**, which is the set of all the possible annotations, so that $A \subseteq U_A$.
- $DO = D \cup A$ is a **set of digital objects** and $do \in DO$ is either a document or an annotation. $U_{DO} = U_A \cup U_D$ is a **universe set of digital objects**, so that $DO \subseteq U_{DO}$.

4.2 Expressing the Temporal Dimension Involved by Annotations

The universe sets U_D , U_A , and U_{DO} are abstract sets, since they contain all the possible needed objects, whether they actually exist or not in any given moment; on the other hand, the sets D , A , and DO are tangible sets that contain the objects that already exist in a given moment: if we pick out an element from D , A , or DO we are dealing with a digital object that has been created even before we start working on it; in other words, the element already exists. The D , A , and DO , sets are a sort of *time-variant sets*, since we can add, delete or modify elements of these sets over time. On the other hand, the U_D , U_A , and U_{DO} sets are a sort of *time-invariant sets*, since they already contain every possible object we may need to deal with.

The annotation is the result of an intellectual task performed on an existing digital object and it follows an already existing digital object. Therefore, the annotation comprises a temporal dimension, which is often not explicit, but that

limits the creation of the annotation to the existence of another digital object. This temporal relationship between the annotation and the annotated digital object does not mean that the annotation cannot be considered a stand-alone intellectual task, but it does impose a temporal ordering between the existence of an annotated digital object and its annotation that cannot be overlooked.

In conclusion, we need some mechanism for rendering the time dimension explicit, if necessary. We will illustrate this mechanism by means of some examples, which show some interesting cases; please note that they do not aim to be exhaustive. Although, these examples do make use of the set DO , they have a more general validity.

Creation of a new digital object consists of the following events.

- (1) We start with the set of digital objects at time k : $DO(k)$.
- (2) We create a new digital object: we pick out an element from the universe set of digital objects that does not belong to $DO(k)$: $do \in \overline{DO}(k) \subseteq U_{DO}$.
- (3) We end up with a new set of digital objects at time $k + 1$, which contains the newly created digital object: $DO(k + 1) = (DO(k) \cup \{do\}) \in 2^{U_{DO}}$.

Therefore, we have the following temporal ordering:

$$\begin{array}{c} \text{time } k \qquad \qquad \qquad \text{time } k+1 \\ \overbrace{DO(k)} \text{ --- } \overbrace{do \in \overline{DO}(k)} \text{ --- } \overbrace{DO(k+1) = DO(k) \cup \{do\}} \\ \text{event 1} \qquad \qquad \text{event 2} \qquad \qquad \qquad \text{event 3} \end{array}$$

both events 1 and 2 happen at time k , but at that time the newly created digital object does not yet belong to the set $DO(k)$ of digital objects at time k ; event 3 happens at time $k + 1$ and represents the new set of digital objects that now also contains the newly created and existing digital object.

Deletion of an existing digital object consists of the following events:

- (1) We start with the set of digital objects at time k : $DO(k)$.
- (2) We choose an existing digital object in the set of digital objects at time k : $do \in DO(k)$.
- (3) We end up with a new set of digital objects at time $k + 1$, which no longer contains the previously chosen digital object: $DO(k + 1) = (DO(k) \setminus \{do\}) \in 2^{U_{DO}}$.

Therefore, we have the following temporal ordering:

$$\begin{array}{c} \text{time } k \qquad \qquad \qquad \text{time } k+1 \\ \overbrace{DO(k)} \text{ --- } \overbrace{do \in DO(k)} \text{ --- } \overbrace{DO(k+1) = DO(k) \setminus \{do\}} \\ \text{event 1} \qquad \qquad \text{event 2} \qquad \qquad \qquad \text{event 3} \end{array}$$

both events 1 and 2 happen at time k ; event 3 happens at time $k + 1$ and represents the new set of digital objects, which does not contain the previously existing digital object.

Modification of an existing digital object consists of the following events:

- We start with the set of digital objects at time k : $DO(k)$.
- We choose an existing digital object in the set of digital objects at time k : $do \in DO(k)$.
- We choose a new digital object, which is the modified version of the previously chosen digital object and does not belong to $DO(k)$: $do' \in \overline{DO}(k) \subseteq U_{DO}$.
- We end up with a new set of digital objects at time $k + 1$, which contains the modified version of the digital object: $DO(k + 1) = (DO(k) \setminus \{do\} \cup \{do'\}) \in 2^{U_{DO}}$.

Therefore, we have the following temporal ordering:

$$\begin{array}{c}
 \overbrace{\hspace{10em}}^{\text{time } k} \qquad \qquad \qquad \overbrace{\hspace{10em}}^{\text{time } k+1} \\
 \underbrace{DO(k)}_{\text{event 1}} \dashrightarrow \underbrace{do \in DO(k)}_{\text{event 2}} \dashrightarrow \underbrace{do' \in \overline{DO}(k)}_{\text{event 3}} \dashrightarrow \underbrace{DO(k+1) = DO(k) \setminus \{do\} \cup \{do'\}}_{\text{event 4}};
 \end{array}$$

events 1, 2, and 3 happen at time k , but at that time the modified digital object does not yet belong to the set $DO(k)$ of digital objects at time k ; event 4 happens at time $k + 1$ and represents the new set of digital objects, which now contains the modified digital object.

These three basic examples reveal our strategy for addressing the time dimension:

- Time k : identify an initial set $DO(k)$ to work with.
- Time k : identify the digital objects to work with, which may belong to $DO(k)$ or not. If the identified digital objects belong to $DO(k)$, then they already exist; on the other hand, if the identified digital objects do not belong to $DO(k)$, then they do not yet exist and therefore this step represents their creation.
- Time $k + 1$: identify the new set $DO(k + 1)$, which results from performing the appropriate operations on the set and the digital objects previously identified.

In all of the cases, both $DO(k)$ and $DO(k + 1)$ contain only digital objects that already exist: this mechanism allows us to unambiguously state which objects we are dealing with in any given moment and the moment when they are able to be utilized.

$DO(k)$ and $DO(k + 1)$ unambiguously identify the digital objects we are dealing with, which are given by $DO(k) \Delta DO(k + 1)$. In particular, the deleted digital objects are given by $DO(k) \setminus DO(k + 1)$ and the newly created digital objects are given by $DO(k + 1) \setminus DO(k)$. Therefore, we can talk about the digital objects identified by the transition from $DO(k)$ to $DO(k + 1)$. We assume that the operations previously shown are atomic: no operation can occur during the execution of another operation, so as to avoid concurrency issues.

In conclusion, this mechanism provides us with a means to clearly identify which objects are involved in a given operation, when they can be utilized, and the ordering among the different events involved by an operation.

The definitions of *collection C* and *repository* provided by [Gonçalves et al. 2004a, p. 295] to a certain extent resemble the proposed mechanism. Indeed, they model the storing of a digital object in the repository as the transition from the original repository to a repository which manages a collection augmented by the newly inserted digital object; similarly, the deletion of a digital object is modeled as the transition from the original repository to a repository which manages a smaller collection without the deleted digital object. On the other hand, the definitions introduced by Gonçalves et al. [2004a] do not explicitly take into account and model the temporal dimension involved with these operations. In this sense, we can consider our mechanism an extension of the proposal made by Gonçalves et al. [2004a].

Furthermore, our mechanism extends, formalizes, and makes more explicit what Rigaux and Spyrtatos [2004, p. 421] left implicit when they said: “in order to define a document formally, we assume the existence of a countably infinite set \mathcal{D} whose elements are used by all authors for identifying the created documents . . . in fact, we assume that the creation of a document is tantamount to choosing a new element from \mathcal{D} .” Indeed, the set \mathcal{D} used by Rigaux and Spyrtatos [2004] corresponds to the set U_{DO} of Definition 4.1 and the creation of a document corresponds to the transition from $DO(k)$ to $DO(k + 1)$, where an object from U_{DO} is chosen and is added to $DO(k)$, thus creating $DO(k + 1)$, as explained above. Furthermore, we provide a formal mechanism for describing the deletion and the modification of a digital object, as well.

In the following sections, we will use the notation $DO(k)$, which explicitly points out the time dimension, only when needed; otherwise we will use the simpler notation DO , without explicitly pointing out the time dimension. We will also use a similar notation for the other sets we will define in the following.

4.3 Handle

According to the previous discussion, we can assume that each digital object is identified by a *unique handle*, which is a name assigned to a digital object to identify and to facilitate the referencing process to the digital object.

Over the past years, various syntaxes, mechanisms, and systems have been developed to provide handles or identifiers for digital objects:

- *Uniform Resource Identifier (URI)* is a compact string of characters for identifying an abstract or physical resource [Berners-Lee 1994b; Kunze 1995; Berners-Lee et al. 1998; Mealling and Denenberg 2002]. The term URL refers to the subset of URIs that identify resources via a representation of their primary access mechanism (e.g., their network “location”), rather than identifying the resource by name or by some other attribute(s) of that resource. The term *Uniform Resource Name (URN)* refers to the subset of URIs that are required to remain globally unique and persistent even when the resource ceases to exist or becomes unavailable [Berners-Lee et al. 1998].
- *Digital Object Identifier (DOI)* is a system that provides a mechanism to interoperably identify and exchange intellectual property in the digital environment. DOI conforms to a URI and provides an extensible framework

for managing intellectual content based on proven standards of digital object architecture and intellectual property management. Furthermore, it is an open system based on non-proprietary standards [Paskin 2006].

- OpenURL aims at standardizing the construction of “packages of information” and the methods by which they may be transported over networks [NISO 2005]. Therefore, OpenURL is a standard syntax for transporting information (metadata and identifiers) about one or multiple resources within URLs, i.e. it provides a syntax for encoding metadata and identifiers, limited to the world of URLs [Paskin 2006].
- Persistent URL (PURL)*¹²: instead of pointing directly to the location of an Internet resource, a PURL points to an intermediate resolution service that associates the PURL with the actual URL and returns that URL to the client as a standard HTTP redirect. The client can then complete the URL transaction in the normal fashion.
- PURL-based Object Identifier (POI)*¹³ is a simple specification for resource identifiers based on the PURL system and closely related to the use of the *Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH)* defined by the *Open Archives Initiative (OAI)*¹⁴ [OAI 2004]. The POI is a relatively persistent identifier for resources that are described by metadata “items” in OAI-compliant repositories.

The following definition introduces the notion of handle, compatible with the mechanisms described above, and its relationship with digital objects.

Definition 4.2. H is a **set of handles** such that $|H| = |DO|$ and $h \in H$ is a generic **handle**. U_H is a **universe set of handles**, which is the set of all the possible handles, such that $|U_H| = |U_{DO}|$; it follows that $H \subseteq U_H$.

We define a bijective function $h : U_H \rightarrow U_{DO}$, which maps a handle to the digital object identified by it:

$$\forall do \in U_{DO}, \exists! h \in U_H \mid h(h) = do \Rightarrow h^{-1}(do) = h.$$

The relationship between the sets H and U_H is the same as the one between the sets DO and U_{DO} , described in section 4.2.

5. COOPERATION

In order to provide users with annotations as an effective cooperation tool, we need to proceed as follows: first, we need to define the notion of user, group of users, and author, as described in Section 5.1; then, we have to deal with both scopes of annotation, as explained in Section 5.3, and various access permissions, as introduced in Section 5.2.

The following definitions do not aim at building a new authentication and authorization scheme from scratch, since many such schemes already exist.

¹²<http://purl.oclc.org/>

¹³<http://www.ukoln.ac.uk/distributed-systems/poi/>

¹⁴<http://www.openarchives.org/>

On the contrary, since a formal model needs to be self-contained and coherent, they aim at providing users of this model with a basic infrastructure for plugging into their preferred authentication and authorization scheme, perhaps extending this model with the additional concepts peculiar to the chosen scheme.

5.1 User, Group of Users and Author

Definition 5.1. Let USR be a **set of users** and $usr \in USR$ is a generic user; U_{USR} is a **universe set of users**, which is the set of all the possible users, so that $USR \subseteq U_{USR}$.

$GR \subseteq 2^{USR}$ is a **set of groups of users** and $G \in GR$ is a generic group of users; $U_{GR} = 2^{U_{USR}}$ is a **universe set of groups of users**, which is the set of all the possible groups of users, so that $GR \subseteq U_{GR}$.

We define a function $gr : USR \rightarrow 2^{GR}$, which maps a user to the groups of users he belongs to. The following constraint must be adhered to:

$$\forall usr \in USR, gr(usr) \neq \emptyset;$$

each user in USR must belong to at least one group of users.

The relationship between the sets USR and GR and the sets U_{USR} and U_{GR} is the same as the one between the sets DO and U_{DO} , described in Section 4.2.

Note that the constraint on the gr function can be also expressed as $\forall usr \in USR, \exists G \in GR \mid usr \in G$. Obviously, a user may belong to more than one group of users, since GR is a subset of the power set of USR or, equivalently, since $gr(usr)$ is an element of the power set of GR .

Digital objects—both documents and annotations—always have at least one author who authored them. Therefore, the author is a specialization of the more general concept of user, introduced in the definition above: an author is a user who authored one or more digital objects.

Definition 5.2. Let us define a function, $au : USR \rightarrow 2^H$, which maps a user to the handles of the digital objects authored by him. Let the **set of authors** AU be the following set:

$$AU = \{usr \in USR \mid au(usr) \neq \emptyset\};$$

we denote with $au \in AU \subseteq USR$ a generic author. The following constraint must be adhered to:

$$\forall h \in H \exists au \in AU \mid h \in au(au);$$

each digital object must be authored by at least one author.

The function au characterizes the authors, distinguishing them from generic users; indeed, if a generic user $usr \in USR$ has not authored any digital object, it follows that $au(usr) = \emptyset$ and thus $usr \notin AU$.

In general, a digital object may have more than one author; in other words, there may exist $au_1, au_2 \in AU \mid au(au_1) \cap au(au_2) \neq \emptyset$. On the other hand, an

author may author more than one digital object, since $au(au)$ is an element of the power set of H .

These definitions allow us to organize the users according to their roles and qualifications. Consider, for example, the case of a genomic annotation that relies on specially trained biologists whose job it is to create highly valuable and domain-specific annotations: these biologists may be users who belong to a group of curators, which grants them the necessary access rights for carrying out their curatorial role.

Finally, note that these proposed definitions share similarities with the notion of society proposed by Gonçalves et al. [2004a, p. 275], when they say “a society is a set of entities and the relationships between them.” The above definitions simply introduce the “set of entities” that come into play plus the distinction between users and authors, which represents a first kind of “relationship” between those entities. Furthermore, the definitions in the following sections will enable the introduction of more “relationships” between users and groups, especially with respect to the access management issue, which is part of what Gonçalves et al. [2004a, p. 276] call “societal governance”.

5.2 Permission

As discussed in Section 3, an annotation can have different access permissions:

- Denied*, if no access is allowed.
- Read only*, if it can be accessed only for reading.
- Read and write*, if it can be read, modified, and deleted.

Definition 5.3. Let $P = \{\text{Denied}, \text{ReadOnly}, \text{ReadWrite}\}$ be a **set of access permissions** and $p \in P$ is an access permission. Let us define the following relations:

—**equality relation** =

$$\{(p, p) \in P \times P \mid p \in P\} = \{(\text{Denied}, \text{Denied}), (\text{ReadOnly}, \text{ReadOnly}), (\text{ReadWrite}, \text{ReadWrite})\}.$$

—**strict ordering relation** $<$

$$\{(\text{Denied}, \text{ReadOnly}), (\text{Denied}, \text{ReadWrite}), (\text{ReadOnly}, \text{ReadWrite})\}$$

—**ordering relation** \preceq

$$\{(p_1, p_2) \in P \times P \mid p_1 = p_2 \vee p_1 < p_2\}.$$

In contrast to the set of the previous definitions, the set of access permissions P is a time-invariant set, which does not need the notation for taking into account the temporal dimension. Indeed, we assume that an annotation can only have the access permissions previously listed. Note that (P, \preceq) is a totally ordered set.

5.3 Scope

As discussed in Section 3, an annotation can have one of the following scopes, which are mutually exclusive:

- Private*, when it can be accessed only by its own author.
- Shared*, when it can be accessed only by a desired set of groups of users.
- Public*, when it can be accessed by all users.

The different scopes of an annotation can be partnered with the various access permissions introduced here, in order to obtain the necessary access policies. For example, in the case of a public annotation we might decide that every user holds read only access to it, while its author has read- and write-access to it. In addition to the policy just described, we might also add a list of groups of users, each one with its specific access permission, in order to pinpoint fine-grained access permission even in the context of a public annotation. Therefore, the proposed formal model does not aim at describing a specific access policy, but at providing the means for describing the different access policies one might need to deal with.

Definition 5.4. Let $SP = \{\text{Private}, \text{Shared}, \text{Public}\}$ be a **set of scopes** and $sp \in SP$ is a scope. Let us define the following relations:

—**Equality relation** =

$$\{(sp, sp) \in SP \times SP \mid sp \in SP\} = \{(\text{Private}, \text{Private}), (\text{Shared}, \text{Shared}), (\text{Public}, \text{Public})\}.$$

—**Strict ordering relation** $<$

$$\{(\text{Private}, \text{Shared}), (\text{Private}, \text{Public}), (\text{Shared}, \text{Public})\}$$

—**Ordering relation** \preceq

$$\{(sp_1, sp_2) \in SP \times SP \mid sp_1 = sp_2 \vee sp_1 < sp_2\}.$$

As in the case of the set of access permissions, the set of scopes SP is also a time-invariant set, because we assume that an annotation can have only one of the three scopes listed above. Note that (SP, \preceq) is a totally ordered set.

6. LINKING

In order to link annotations to digital objects and to correctly anchor annotations to digital objects, we need to proceed as follows: first, we need to choose a linking mechanism and define the link types that can exist between annotations and digital objects, as described in Section 6.1; then, since annotations are usually linked to specific parts of a digital object, we need to model the content of digital objects, as explained in Section 6.2; finally, a suitable anchoring mechanism for annotations has to be provided, as introduced in Section 6.3.

6.1 Linking Annotations to Digital Objects

Handles can be used not only for the purpose of uniquely identifying a digital object, but they can also provide us with a means for linking an annotation to a digital object. This use of handles is particularly clear if we think about URLs, but it is also still valid in the case of the other types of handles presented in Section 4.3.

Once we have decided to use handles as a basic mechanism for linking annotations to digital objects, we still have to consider the kind of links an annotation can have with a digital object. Annotations can be linked to digital objects with two main types of links:

—*Annotate link*: an annotation annotates a digital object, which can be a document or another annotation.

The “annotate link” is intended to allow an annotation to only annotate one or more parts of a given digital object. Therefore, this kind of link lets the annotation express *intra-digital object relationships*, meaning that the annotation creates a relationship among the different parts of the annotated digital object;

—*Relate-to link*: an annotation relates to a digital object, which can be a document or another annotation.

The “relate-to link” is intended to allow an annotation to only relate to one or more parts of other digital objects, but not the annotated one. Therefore, this kind of link lets the annotation express *inter-digital object relationships*, meaning that the annotation creates a relationship between the annotated digital object and the other digital objects related to it.

With respect to these two main types of link, we introduce the following constraint:

an annotation must annotate one and only one digital object, which can be either a document or another annotation—an annotation must have one and only one “annotate link.”

This constraint means that an annotation can be created only for the purpose of annotating a digital object and not exclusively for relating to a digital object. An annotation then, can annotate one and only one digital object, because the “annotate link” expresses *intra-digital object relationships* and thus it cannot be mutual to multiple digital objects different from the annotated one. Finally, this constraint does not prevent the annotation from relating to more than one digital object: from having more than one “relate-to link.”

This situation is very similar to what happens in the real world. When we deal with paper documents, we can *annotate* one or more parts of the document that we have at hand; this document also provides us with the physical medium for writing the content of the annotation. On the other hand, the content of the annotation can contain references to other documents; in other words, it can *relate* the document at hand to other documents that are currently being viewed. Therefore the act of annotating concerns one and only one document, to which

the annotation is anchored, although there may be one or more references that relate the annotation to other documents. One could argue that in the digital world these limitations could be overcome and that an annotation could annotate multiple documents at the same time. Apart from being possible, what could we gain from this option? If we allow multiple annotated links, we are going to add some uncertainty because the annotation would lose its strong relationships with only one object. In fact, this object represents its main purpose, while linking the annotation to multiple objects would give us unclear semantics.

Therefore, we opt for constraining the link types that an annotation can have, and the following definition introduces the set of allowed link types.

Definition 6.1. Let LT be a **set of link types**; an element $lt \in LT$ corresponds to one of the allowed link types. The set LT contains the following link types: $LT = \{\text{Annotate}, \text{RelateTo}\}$.

As in the case of the set of access permissions and the set of scopes, the set of link types LT is also a time-invariant set, because we assume that an annotation can be linked to digital objects only with the link types listed here.

6.2 Stream

Digital objects can be very different—texts, images, audio, videos, hypertexts, multimedia objects, and so on—and the way in which their structure and content is modeled and expressed can also widely vary across different conceptual and logical models of DL and digital object. Nevertheless, many of these types of models share the idea that beyond representing the structure of the digital object, the model also has to take into account a mechanism for representing the actual content of the digital object. For example, Navarro and Baeza-Yates [1997] and Gonçalves et al. [2004a] both use the notion of *stream*, which is an ordered sequence of symbols representing the actual content of a digital object or part of it. Bottoni et al. [2003] define the content of a digital object as a function from a set of indices to a set representing the vocabulary of the symbols. Finally, Castelli and Pagano [2002a, 2002b] associate each digital object with many different *manifestation* entities, which represent sequences of bytes and files containing different parts of the digital object itself.

The following definition introduces the concept of stream in order to represent the actual content of a digital object or a part of it. The definition of stream is inspired by Navarro and Baeza-Yates [1997], and Gonçalves et al. [2004a] but with some differences, which will be discussed in the following.

Definition 6.2. A **stream** sm is a finite sequence:

$$sm : I = \{1, 2, \dots, n\} \rightarrow \Sigma, \quad n \in \mathbb{N},$$

where Σ is the *alphabet of symbols*. We allow the existence of an **empty stream** $esm = \emptyset$. SM is a **set of streams** and $sm \in SM$ is a stream. U_{SM} is a **universe set of streams**, that is, the set of all the possible streams. It follows that $SM \subseteq U_{SM}$.

We define a function $\text{hsm} : H \rightarrow 2^{SM}$, which maps a handle of a digital object to the streams contained in that digital object. The following constraint must be adhered to:

$$\forall h \in H, \text{hsm}(h) \neq \emptyset.$$

Each digital object must contain at least one stream, which could also possibly be the empty stream.

The relationship between the sets SM and U_{SM} is the same as the one between the sets DO and U_{DO} , described in Section 4.2.

The stream is required to be neither a surjective nor an injective function. We can exploit the non surjectivity of the stream in order to use standard sets—characters, numbers, and so on—as a codomain for a stream; otherwise, if the function were constrained to be surjective, we would be forced to use an “ad hoc” codomain for each different stream. On the other hand, since the stream is not an injective function, it is therefore not invertible: in fact, when given a symbol, we cannot trace this symbol back to its position within the stream.

For example, if we consider the following piece of text:

1 2 3 4 5 6 7 8 9 10

T i t l e □ T e x t,

then we can define the stream

$$\text{sm} : I = \{1, 2, \dots, 10\} \rightarrow \Sigma = \{A, B, \dots, Z, a, b, \dots, z, \sqcup\},$$

such that $\text{sm}(1) = \text{T}$, $\text{sm}(2) = \text{i}$, \dots , $\text{sm}(10) = \text{t}$. Note that if the stream was constrained to be surjective, we should use a codomain Σ' constituted only by the letters of the piece of text shown above: $\Sigma' = \{\text{T}, \text{i}, \text{t}, \text{l}, \text{e}, \sqcup, \text{x}\}$. In any case, from a given symbol, for example “t,” we cannot unambiguously determine its position in the stream, because the stream is not injective—“t” is given by both $\text{sm}(3)$ and $\text{sm}(10)$.

In particular, we can distinguish two main kinds of streams:

- Logical stream** lsm : this is a stream in which each element $\sigma \in \Sigma$ represents a *logical symbol* within the stream.
- Physical stream** psm : this is a stream in which each element $\sigma \in \Sigma$ represents a *physical symbol* within the stream.

Now we will discuss the distinction between logical and physical streams by means of an example. Although the map shown between natural numbers and letters is quite intuitive, it should be pointed out that the elements of the set Σ are symbols that abstract the underlying encoding of the text. For example, if we consider an ASCII text, each element of the set Σ corresponds to exactly one byte in the physical text stream; thus, we should use the codomain

$$\Sigma_{\text{ASCII}} = \{41_{16}, 42_{16}, \dots, 5A_{16}, 61_{16}, 62_{16}, \dots, 7A_{16}, 20_{16}\}$$

instead of Σ to represent the actual stream. On the other hand, in the case of a UNICODE text, each element of the set Σ corresponds to two bytes in the

physical text stream; thus, we should use the codomain:

$$\Sigma_{\text{UNICODE}} = \{00_{16}41_{16}, 00_{16}42_{16}, \dots, 00_{16}5A_{16}, 00_{16}61_{16}, 00_{16}62_{16}, \dots, \\ 00_{16}7A_{16}, 00_{16}20_{16}\}$$

instead of Σ . In this latter case we are forced to define the elements of Σ_{UNICODE} as two-byte pairs in order to map the indices of I into the symbols of Σ_{UNICODE} ; as a result, we would not be able to access each byte individually. If we wish to access each byte of the UNICODE stream, we should define the following domain $I' = \{1, 2, \dots, 20\}$ and codomain

$$\Sigma'_{\text{UNICODE}} = \{00_{16}, 41_{16}, 42_{16}, \dots, 5A_{16}, 61_{16}, 62_{16}, \dots, 7A_{16}, 20_{16}\}$$

for the stream; but in this case we would lose the correspondence with the ten letters of the piece of text, because two indices in I' would correspond to each letter of the piece of text. This example demonstrates that on the one hand we have a logical stream, which represents the piece of text, and on the other there are one or more physical streams that represent the physical encoding of the piece of text shown. Similar, and even more complex, considerations can be made in the case of audio, image, and video streams, where the complexity of such streams increases the choices available for representing them both in logical and in physical terms. Another example is the compression of streams, where more symbols in one stream correspond to fewer symbols in the other stream.

This observation points out the need to carefully define the level of abstraction of a stream and the degree of detail that need to be adopted when defining streams. In other words, should we model the physical encoding of a stream, or some more abstract representation of that stream? Depending on the case, both levels of abstraction may be necessary: for example, when we do a macro-comparison of two digital libraries, we can use more abstract streams; on the other hand, if we want to precisely describe the functioning of some component of a digital library, as a repository, we need to use streams that better represent the physical encoding of the objects in the repository. However, past experience in the field of DBMS teaches us that it is better to keep the logical and the physical levels distinct. This is why we want to distinguish between logical and physical streams.

Note that Navarro and Baeza-Yates [1997] make use of logical streams, but they do not specify much about physical streams, leaving them to the implementation of the system. Neither Bottoni et al. [2003] nor Gonçalves et al. [2004a] addressed this problem at all, but in Gonçalves and Fox [2002] it turns out that streams are essentially identified by *Multipurpose Internet Mail Extensions (MIME)* types [Freed and Borenstein 1996a, 1996a, 1996b; Moore 1996; Freed et al. 1996] and thus, they are substantially physical streams. Finally, since the notion of *manifestation* used by Castelli and Pagano [2002b, 2002a] refers to a physical file holding part of the content of a digital object, these authors essentially use physical streams also, and in a way that resembles the implementation of streams by Gonçalves and Fox [2002].

We believe that the research field of digital libraries also needs to clearly distinguish between the logical and physical levels; this distinction is a

prerequisite for each formal model of DL that aims to be sufficiently clear, expressive and flexible. Moreover, an explicit and formal mechanism for modeling the relationship between logical and physical streams, and the properties of such a relationship is needed. This will be the focus of the next definition.

Definition 6.3. Given two streams $sm_1, sm_2 \in SM$, let us define a **stream mapping relation**:

$$SMR(sm_1, sm_2) = \{(i, j) \in I_{sm_1} \times I_{sm_2} \mid sm_1(i) \text{ is mapped to } sm_2(j)\}.$$

Let us define a **stream mapping set** $SMS = \{SMR(sm_1, sm_2)_i\}$ such that:

- (1) $\forall sm \in SM, \exists SMR \in SMS \mid SMR(sm, sm) = \{(i, j) \in I_{sm} \times I_{sm} \mid i = j\}$.
- (2) $\forall SMR(sm_1, sm_2) \in SMS, \exists SMR(sm_2, sm_1) \in SMS \mid$
 $SMR(sm_2, sm_1) = SMR^{-1}(sm_1, sm_2) =$
 $\{(j, i) \in I_{sm_2} \times I_{sm_1} \mid (i, j) \in SMR(sm_1, sm_2)\}$.
- (3) $\forall SMR(sm_1, sm_2), SMR(sm_2, sm_3) \in SMS, \exists SMR(sm_1, sm_3) \in SMS \mid$
 $SMR(sm_1, sm_3) = SMR(sm_1, sm_2) \circ SMR(sm_2, sm_3) =$
 $\{(i, k) \in I_{sm_1} \times I_{sm_3} \mid (i, j) \in SMR(sm_1, sm_2) \wedge (j, k) \in SMR(sm_2, sm_3)\}$.

Let us define a **stream mapping set indicator function**:

$$\chi_{SMS}(sm_1, sm_2) = \begin{cases} 1 & \text{if } SMR(sm_1, sm_2) \in SMS \\ 0 & \text{if } SMR(sm_1, sm_2) \notin SMS. \end{cases}$$

Each element $(i, j) \in SMR(sm_1, sm_2)$ represents the fact that the i -th symbol in the first stream is related to the j -th symbol in the second stream. The SMR relation represents and embeds the algorithm that allows us to map symbols of one stream into those of the other. In particular, in the case of the relationship between logical and physical streams, the SMR relation represents the fact that given logical symbols are encoded with given physical symbols. In this way, it clearly models the distinction and the passage from the logical to the physical level. For example, the SMR relation could represent the mapping between the pixels of an image and its *Joint Photographic Experts Group (JPEG)* encoding.

In general, the stream mapping relation allows us to express many-to-many relationships between symbols of two streams. In particular, we are interested in expressing, at least, the following relationships:

- A one-to-one relationship between the symbols of the two streams, as in the previous example of the piece of text, and its ASCII encoding.
- A one-to-many relationship between the symbols of the two streams, as in the previous example of the piece of text and its UNICODE encoding.
- A many-to-one relationship between the symbols of the two streams, as in the case of compression of one stream into another.

The stream mapping relation provides us with a further degree of freedom, since we can have symbols in a stream that do not correspond to another stream. In this way, we can model some kind of loss of information due to different encodings.

Finally, the stream mapping relation enables the same logical symbol to be encoded in different ways according to its position in the stream. Consider the letter “t,” which in the previous example appears in the third and tenth position of the stream; it could be encoded in two different ways if we apply some compression algorithm to that stream.

Definition 6.3 allows us to associate a set of physical streams to the same logical stream, providing us with a mechanism to enable different encodings of the same logical stream. We could also create a *chain of streams*: we could specify that a logical stream is encoded with a given physical stream, and that this physical stream is mapped to another physical stream and so on. These observations led us to introduce the stream mapping set, which contains the stream mapping relations and holds the intuitive and expected properties for this kind of set:

- Reflexive*: for each stream, the obvious mapping of the stream to itself exists.
- Symmetric*: if we know how to map one stream to another, we can also map the second stream back to the first one.
- Transitive*: if we know the mapping between one stream and another and we also know the mapping between the second stream and a third, we know how to map the first to the third.

Now we can study the impact Definition 6.3 has on the set of streams SM and how it contributes to enforcing the distinction between the logical and the physical levels.

PROPOSITION 6.4. *The following relation:*

$$SMS = \{(sm_1, sm_2) \in SM \times SM \mid \chi_{SMS}(sm_1, sm_2) = 1\}$$

is an equivalence relation on the set of streams SM . The sets:

$$SM = \{sm_1, sm_2 \in SM \mid (sm_1, sm_2) \in SMS\}$$

are the equivalence classes of all the streams of SM that are mapped one to another, and SM/SM is the quotient set.

PROOF. The relation is:

- Reflexive*: $\forall sm \in SM, \exists SM\mathcal{R}(sm, sm) \in SMS \Rightarrow \chi_{SMS}(sm, sm) = 1$.
- Symmetric*: $\forall sm_1, sm_2 \in SM \mid \chi_{SMS}(sm_1, sm_2) = 1, \exists SM\mathcal{R}(sm_2, sm_1) \in SMS \Rightarrow \chi_{SMS}(sm_2, sm_1) = 1$.
- Transitive*: $\forall sm_1, sm_2, sm_3 \in SM \mid \chi_{SMS}(sm_1, sm_2) = \chi_{SMS}(sm_2, sm_3) = 1, \exists SM\mathcal{R}(sm_1, sm_3) \in SMS \Rightarrow \chi_{SMS}(sm_1, sm_3) = 1$.

Therefore, it is an equivalence relation. \square

We can choose the logical stream as representative of the equivalence class; the SMS equivalence relation allows us to deal only with logical streams. In fact, it removes us from the physical level and hides the details of the representation and the encoding of logical streams into physical ones. Therefore, the SMS equivalence relation enforces the distinction between the logical and the

physical levels and provides us with the means of working and reasoning at a logical level, clearly separating it from the physical one.

Furthermore, we can iterate this line of reasoning and use this equivalence relation as a basic mechanism for introducing further levels of abstraction and creating a kind of hierarchy among streams. Indeed, on the quotient set SM/SM we could introduce an equivalence relation similar to SM/S in order to express the fact that two or more logical streams can be mapped to each other. This is how we can maintain more abstract classes of equivalent logical streams on the quotient set SM/SM , by keeping them distinct from the different ways in which they can be encoded; this different encoding is, in turn, represented by the less abstract equivalence classes on the set SM . This procedure can be repeated as many times as needed in relation to the number of levels of abstraction. For example, suppose we have a piece of text that can be represented either as a sequence of characters or as a scanned image. These are two different logical streams, that can be encoded with many different physical streams. In this case, a first level of abstraction is to put all the physical streams that encode the character streams into one equivalence class created on SM , and all the physical streams that encode the image stream into another equivalence class created on SM . However, a higher level of abstraction is to put both the equivalence class of the text stream and the equivalence class of the image stream into a new and more abstract equivalence class created on SM/SM , in order to express the fact that both of them are representations of the same piece of text.

For all these reasons, Definition 6.3 and Proposition 6.4 constitute a step forward with respect to previous models [Castelli and Pagano [2002a, 2002b] Gonçalves et al. 2004a, 2004b; Navarro and Baeza-Yates 1997; Bottoni et al. 2003], which only partially address this issue or do not address it at all. On the other hand, Definitions 6.2, 6.3, and Proposition 6.4 are fully compatible with the definition of stream provided by both Navarro and Baeza-Yates [1997] and Gonçalves et al. [2004a]; thus, we can utilize the proposed distinction between logical and physical streams in both the models provided by Navarro and Baeza-Yates [1997] and Gonçalves et al. [2004a] in order to extend such models, if necessary.

6.3 Segment

The handles discussed in Section 4.3 may be capable not only of uniquely identifying a digital object, but also of indicating a part of the identified digital object. For example, a URL can point to any given anchor within a HTML document, or we can use an XPath expression to point to a specific element within an XML document. On the other hand, parts of a digital object cannot always be identified with an arbitrary degree of detail; for example, a URL cannot point to a given word of a HTML document, if this word is not marked with an anchor. Therefore, we need some further mechanism for identifying parts of a digital object with the necessary degree of detail.

The following definition introduces the notion of *segment*, which is a mechanism for selecting parts of a stream; this mechanism can be partnered with the

handle of a digital object to provide access to a digital object with the necessary degree of detail.

Definition 6.5. Given a stream $sm : I = \{1, 2, \dots, n\} \rightarrow \Sigma$, $n \in \mathbb{N}$, $sm \in SM$, a **segment** is a pair:

$$st_{sm} = (a, b) \mid 1 \leq a \leq b \leq n, \quad a, b \in \mathbb{N}.$$

A **stream segment** is a restriction, $sm_{|[a,b]}$, of the stream sm to interval $[a, b]$ associated with the segment st_{sm} . ST is a **set of segments** and $st_{sm} \in ST$ is a generic segment; U_{ST} is a **universe set of segments**, which is the set of all the possible segments, so that $ST \subseteq U_{ST}$.

The relationship between the sets ST and U_{ST} is the same as the relationship between the sets DO and U_{DO} , described in Section 4.2. Definition 6.5 resembles the definition of segment provided in Navarro and Baeza-Yates [1997], and Gonçalves et al. [2004a].

We can assume that logically related symbols of logical streams are contiguous and are in an ascending order. This assumption goes well with Definition 6.5, which selects a series of contiguous symbols. On the other hand, Definition 6.3 allows us to disregard this constraint for the mapping to physical streams, since the stream mapping relation \mathcal{SMR} allows us to map the contiguous symbols of the logical stream to noncontiguous symbols of the physical stream. For example, the indices $I_{sm} = \{1, 2, 3, 4, 5\}$ of a logical stream could be mapped to the indices $I_{psm} = \{13, 7, 19, 9, 15\}$ of a physical stream. If we choose the segment $st_{lsm} = (2, 4)$, which is associated with the interval $[2, 4]$ for the logical stream, we are not forced to map it to the interval $[7, 9]$, obtained by mapping the segment st_{lsm} to a corresponding segment $st_{psm} = (7, 9)$, of the physical stream. On the contrary, we can map each index in the interval $[2, 4]$ to its corresponding index in the physical stream, obtaining the set of indices $\{7, 19, 9\}$, which do not fit in the interval $[7, 9]$. See Navarro and Baeza-Yates [1997] for further explanation about ordering in multimedia streams.

This feature is important because symbols that are contiguous in a logical stream can correspond to non-contiguous symbols in a physical stream, due to some kind of compression, for example. In addition, Proposition 6.4 allows us to reason only in terms of logical streams that comply with the assumption made above, without worrying about the physical streams that are in the same equivalence class of the logical stream. This observation further highlights the benefits that may arise by clearly distinguishing between the logical and the physical levels.

All of the introduced concepts, namely handle, stream, and segment, provide us with the formal means needed to deal with the linking and anchoring problem related to annotations. By using a *handle* h we can link an annotation to a digital object; then, the function $hsm(h)$ allows us to select the desired *stream* sm of the digital object identified by h , be it a physical or a logical view of the actual content of the digital object; finally, a *segment* st_{sm} enables the fine-tuned anchoring of the annotation to the digital object.

Last, we can also rely on these concepts to address the annotation repositioning problem that arises when the content of annotated digital objects changes.

Indeed, the introduced concepts offer us the possibility of modeling what Phelps and Wilensky [2000b] call *robust location*, which are redundant descriptors of locations within a digital object created by using a number of different data records. Moreover, we can also express the algorithms they propose for reattaching annotations to a digital object when the annotated digital object is modified. Furthermore, these concepts can provide us with a common grounding not only for designing repositioning algorithms, as in the case of Phelps and Wilensky [2000b], but also for studying and describing what users expect an annotation system to do when annotated digital objects change, as done in the user study conducted by Brush et al. [2001].

7. MATERIALIZATION

As in Agosti and Ferro [2003a], we define *sign of annotation* as the basic way in which an annotation can take shape, the way of representing and materializing the semantics of annotation. For example, we can identify the following basic signs of annotations:

- *Textual sign*: a textual materialization of the semantics of an annotation, which is expressed by a piece of text added to a digital object.
- *Graphic sign*: the graphic materialization of the semantics of an annotation, which is expressed by a graphic mark added to a digital object.
- *Video sign*: the video materialization of the semantics of an annotation, which is expressed by a video fragment added to a digital object.
- *Auditive sign*: the auditive materialization of the semantics of an annotation, which is expressed by an audio fragment added to a digital object.

These basic signs can be combined to express more complex signs of annotation. Consider the example of Figure 2, where two annotations are shown, one in the upper part near the *auditive sign* bullet, and the other in the lower part near the *auditive sign* bullet. The first annotation is constituted by both a basic sign and a compound sign. The highlight is a basic graphic sign, while the call-out is a compound sign. It is in turn formed by two graphic signs, the box and the arrow, and by a textual sign, which is the question “Wouldn’t also it be useful for visually impaired people?”. The second annotation is made by three basic signs: two graphic signs, the arrow and the cloud, and a textual sign, which is the answer “I think so!”.

In conclusion, by using the notion of sign of annotation, we consider the annotation as possibly complex multimedia, constituted by different parts, each one with its own medium. The following definition formally introduces the concept of sign of annotation.

Definition 7.1. A **sign of annotation** is a stream. $SN \subseteq SM$ is a **set of signs of annotation** and $sn \in SN$ is a sign. $U_{SN} \subseteq U_{SM}$ is a **universe set of signs of annotation**, which is the set of all the possible signs of annotation, so that $SN \subseteq U_{SN}$.

The relationship between the sets SN and U_{SN} is the same as the relationship between the sets DO and U_{DO} , described in Section 4.2.

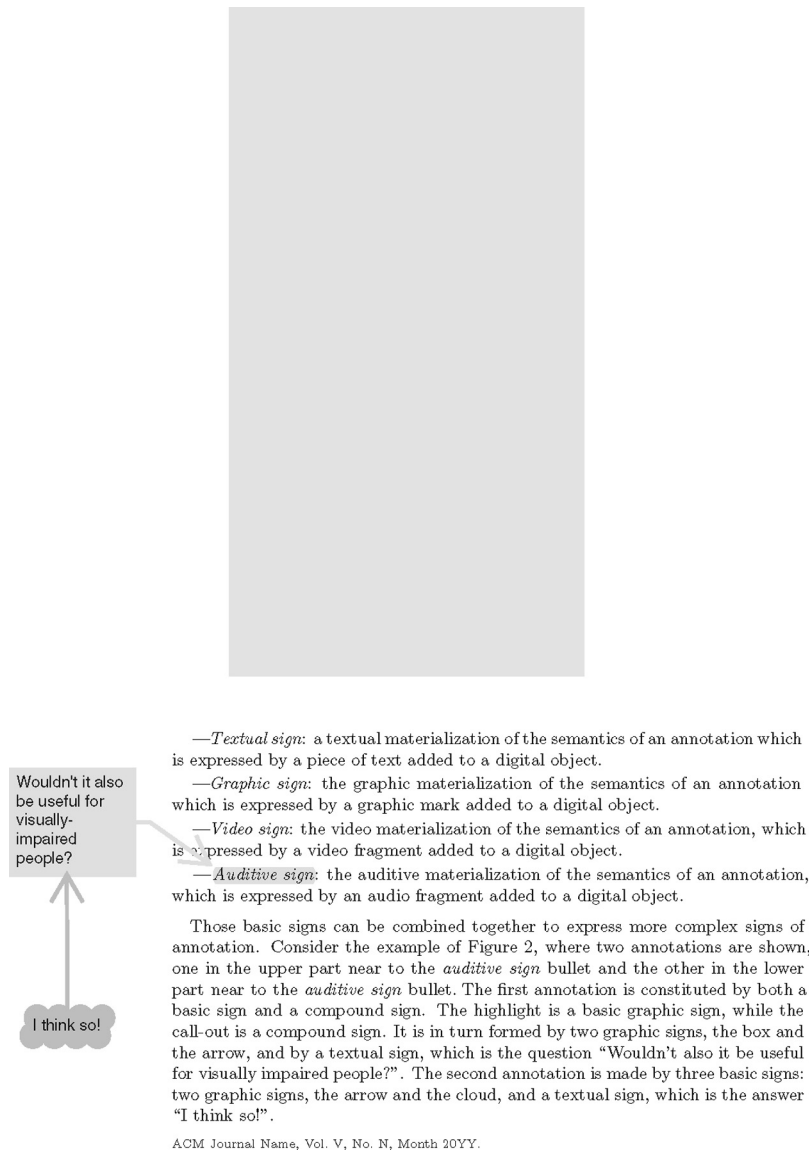


Fig. 2. Compounding of basic signs of annotation.

Henceforth we will use the term *sign of annotation*, or briefly stated as *sign*, to indicate a stream that belongs to an annotation. On the other hand, we will use the term *stream* to indicate a stream that belongs to a digital object without the need of specifying if the digital object is a document or an annotation.

8. SEMANTICS

As in Agosti and Ferro [2003a], we define *meaning of annotation* as a main feature of the concept of annotation, which identifies conceptual differences within

the semantics of the annotation or part of it. For example, looking at the different points of view concerning annotations introduced in Section 2, we can see that they correspond to different and very broad meanings of annotation. Furthermore, we can identify different meanings of annotation within each given viewpoint: for example, within the viewpoint called “annotation as content” we can identify at least the following meanings of annotation, but many others would be possible, also depending on specific domains:

- Comprehension and study*: annotating a document is a way of better investigating and understanding a concept. This process principally involves a *private scope*, because the recipient of an annotation is the person who created it. Other people reading an annotated document may benefit from existing annotations as well.
- Interpretation and elucidation*: annotating a document could be a way of adding comments and explaining sentences within it. The aim is to make it more comprehensible and to exchange ideas on a topic; an example could be an expert in literature who explains and annotates the Divine Comedy. This process principally involves a *public scope*, and the recipients of an annotation are people who are not necessarily related to the creator of the annotation.
- Cooperation and revision*: a team of people could annotate a document for various purposes, as they are working on a common document or they are reviewing someone else’s work; annotating a text is thus a way of sharing ideas and opinions in order to improve a text. This process principally involves a *shared scope*, because the recipient of an annotation is a team of people working together on a given subject.

As a further example, if we consider annotations as metadata, the meaning of the annotation could be provided by some standard metadata specification, such as the ones provided by the DCMI, which deals with the development of interoperable online metadata standards. Last, it is also possible to organize the meanings of annotations according to some kind of hierarchy, such as a taxonomy or an ontology, in order to provide navigation capabilities among different meanings of annotation.

Definition 8.1. M is a **set of meanings of annotations**, and $m \in M$ is a generic **meaning of annotation**.

The **meanings graph** is a labeled directed graph (G_M, l_M) , where $G_M = (M, E_M \subseteq M \times M)$ and $l_M : E_M \rightarrow L_M$ with L_M set of labels.

The **meanings function** $m : SN \rightarrow 2^M$ associates each sign of annotation with its corresponding meanings of annotation. The following constraint must be satisfied:

$$\forall sn \in SN, m(sn) \neq \emptyset;$$

each sign of annotation has at least one meaning of annotation.

As in the case of the set of access permissions, the set of scopes, and the set of link types LT , the set of meanings M is a time-invariant set, because we assume

that meanings represent preexisting knowledge that does not change over time. Therefore, all the needed meanings of annotation are already elements of the set M .

The goal of the meanings graph is to provide structure and hierarchy among the meanings of annotation in order to navigate and browse through them. The relation E_M can be constrained in many ways to obtain the necessary structure of meanings, which can represent some domain specific knowledge. The labelling function l_M can be further exploited to distinguish different kinds of arcs in the set E_M in order to better explain the kind of relationship between two different meanings.

Gonçalves et al. [2004a] introduce the general notion of *structure* in DLs, represented by a labeled directed graph, as a means of expressing different kinds of structure that might be needed in DLs, such as taxonomies, metadata, and so on. Therefore, the meanings graph adheres to this definition of structure, and it is a structure aimed at enabling navigation through the different meanings of annotation.

The meanings function allows us to associate each sign of annotation with its corresponding meanings in order to clarify the semantics of the sign. Note that the meanings function is neither injective nor surjective. In conclusion, an annotation is expressed by one or more signs of annotation, which in turn are characterized by one or more meanings of annotation, thus defining the overall semantics of the annotation.

The explicit distinction between the meaning and the sign of annotation is quite new in the field of annotations. Indeed, annotations are generally typed as a whole object according to some predefined set of annotation types [W3C 2005a; Kahan and Koivunen 2001; Frommholz et al. 2003; Bottoni et al. 2003, 2004], but there is usually no means for describing the semantics of an annotation with the necessary level of precision. However, this is possible with the meanings of annotation. Furthermore, annotation types do not allow any kind of navigation among different types, while meanings of annotations can be organized to do that.

Some helpful information about the choice of distinguishing between meaning and sign of annotation can be obtained from the field of human computer interaction. Bottoni et al. [1999] deal with visual languages and define *Characteristic Structures (CS)* as sets of image pixels forming functional or perceptual units whose recognition results in the association of the CS with a meaning. They call *Characteristic Patterns (CP)* the CS along with descriptions of the CS and a relation that associates descriptions to CS and viceversa. The distinction between CS and CP resembles the distinction between sign and meaning of annotation; Fogli et al. [2004] also recognize this correspondence and say that “an annotation is a complex CS interpreted by a human as a CP”. On the other hand, Bottoni et al. [2003] also adopt the CS and CP mechanism in the context of annotations, but they use this mechanism to place annotations on information resources rather than to distinguish between the semantics and the materialization of annotations.

We are interested in studying the sharing of common meanings among different signs, defined as the basic mechanism for relating and gathering up

different signs that express common semantics. This is very helpful in the case of annotations made by two different users. For example, they may use different signs to indicate the importance of a passage—an asterisk or an exclamation mark; knowing that these two different signs have the same meaning allows these two users to communicate and interact with each other.

In addition, this would also help us to disambiguate cases where two signs that look exactly the same have two different meanings. For example, consider the case of a user who is performing different tasks: while he is studying a paper, he may highlight a passage to indicate that it is worth further investigation; while he is reviewing a paper, he may highlight a passage to indicate that it is not correct. In both cases, he uses the same kind of sign but with two different semantics.

The most immediate way of approaching this issue is to introduce the following relation:

$$\mathcal{M}_1 = \{(sn_1, sn_2) \in SN \times SN \mid m(sn_1) \cap m(sn_2) \neq \emptyset\}.$$

This relation clearly highlights the signs that directly share some common meanings. However, this relation is not able to relate two signs that do not directly share a common meaning.

Therefore, a step forward also considers both the meanings graph $G_M = (M, E_M)$ and its reflexive transitive closure $G_M^* = (M, E_M^*)$, so that we can introduce the following relation:

$$\begin{aligned} \mathcal{M}_2 = \{ & (sn_1, sn_2) \in SN \times SN \mid \exists m \in M, m_1 \in m(sn_1), m_2 \in m(sn_2), \\ & (m_1, m_2) \in E_M^* \vee (m_2, m_1) \in E_M^* \vee \\ & ((m, m_1) \in E_M^* \wedge (m, m_2) \in E_M^*) \vee ((m_1, m) \in E_M^* \wedge (m_2, m) \in E_M^*) \}. \end{aligned}$$

The \mathcal{M}_2 relation means that two signs, s_1 and s_2 , are in relation if among their meanings, which are obtained by $m(sn_1)$ and $m(sn_2)$, at least:

- one is the ancestor of the other $((m_1, m_2) \in E_M^* \vee (m_2, m_1) \in E_M^*)$; or
- they both have a common ancestor $((m, m_1) \in E_M^* \wedge (m, m_2) \in E_M^*)$; or
- they both are the ancestors of a common meaning $((m_1, m) \in E_M^* \wedge (m_2, m) \in E_M^*)$; or
- two meanings are equal—as in the case of the \mathcal{M}_1 relation.

Indeed, $\mathcal{M}_1 \subseteq \mathcal{M}_2$ because $\forall s_1, s_2 \in SN \mid (sn_1, sn_2) \in \mathcal{M}_1 \Leftrightarrow \exists m \in m(sn_1) \cap m(sn_2) \Rightarrow (m, m) \in E_M^* \Leftrightarrow (sn_1, sn_2) \in \mathcal{M}_2$.

\mathcal{M}_2 is a very broad relation that allows us to relate different signs according to the four strategies outlined above. Where needed, we could use limited versions of \mathcal{M}_2 that adopt only some of the strategies introduced here—for example, \mathcal{M}_1 uses only the last strategy.

Further strategies can be envisaged to group signs on the basis of their meanings; for example, we could take into consideration the predecessor of a meaning instead of its ancestor, as in \mathcal{M}_2 . Therefore, the \mathcal{M}_1 and \mathcal{M}_2 relations are examples of the utilization of the meanings graph; however, they are not

intended to be exhaustive. For example, Rigaux and Spyrtatos [2004] propose a subsumption relation on the terms of a taxonomy, and a way of navigating through them that can also be very useful in the context of the meanings of annotation.

9. ANNOTATION

We are now ready to introduce the definition of annotation. Summing up the concepts introduced in the previous sections, we can briefly say that an annotation is expressed by one or more signs of annotation, such as a piece of text or some graphic mark, which are the way an annotation takes shape. The semantics of each sign is, in turn, defined by one or more meanings of annotation. With respect to the linking issue, an annotation must annotate one and only one digital object, identified by its handle, while it may relate to one or more digital objects. Last, the mechanism introduced in Section 4.2 on how to address the time dimension is fundamental to properly define the relationship between the annotation and the annotated digital object.

Definition 9.1. An **annotation** $a \in A(k)$ is a tuple:

$$a = (h_a \in H(k), au_a \in USR(k-1), G_a \in 2^{GR(k-1)} \times P, sp_a \in SP, \\ \mathcal{A}_a \subseteq SN(k) \times LT \times ST(k) \times SM(k-1) \times H(k-1)),$$

where:

- h_a is the unique handle of the annotation a , i.e. $h(h_a) = a$.
- au_a is the author of the annotation a : i.e. $h_a \in au(au_a)$.
- G_a are the groups of users with their respective access permissions for the annotation a , specified by the pairs (G, p) with $G \in G_a$ and $p \in P$.
- sp_a is the scope of the annotation a .
- Each n-ple of the \mathcal{A}_a relation means that the annotation a by means of a sign in $SN(k)$ and a link type in LT is annotating or relating to a segment in $ST(k)$ of a stream in $SM(k-1)$ of a digital object identified by its handle in $H(k-1)$.

Note that since $\forall sm \in SM(k-1) \mid \exists \alpha \in \mathcal{A}_a, \alpha = (sn, t, st_{sm}, sm, h)$ must be $sm \in hsm(h)$; in other words, the stream sm must be contained in the digital object identified by the handle h .

We introduce the following auxiliary sets to simplify the following discussion:

- The set of the signs of annotation that belong to the annotation a :
 $SN_a = \{sn \in SN(k) \mid \exists \alpha \in \mathcal{A}_a, \alpha = (sn, lt, st_{sm}, sm, h)\} = hsm(h_a)$.
- The set of the handles of digital objects that are subject to the tasks of the annotation a :

$$H_a = \{h \in H(k-1) \mid \exists \alpha \in \mathcal{A}_a, \alpha = (sn, lt, st_{sm}, sm, h)\}.$$

The following constraints must be adhered to:

- (1) The annotation a must annotate one and only one digital object, and it cannot also relate to this digital object, hence:

$$\begin{aligned} & \exists! h \in H_a \mid \\ & (\forall \text{sn} \in SN_a, \exists! \alpha \in \mathcal{A}_a, \alpha = (\text{sn}, \text{Annotate}, st_{\text{sm}}, \text{sm}, h)) \wedge \\ & (\nexists \alpha_1 \in \mathcal{A}_a, \alpha_1 = (\text{sn}_1, \text{RelateTo}, st_{\text{sm}_1}, \text{sm}_1, h)). \end{aligned}$$

- (2) A sign in SN_a cannot relate to more than one digital object, hence:

$$\begin{aligned} & \forall \text{sn} \in SN_A \mid \exists \alpha_1, \alpha_2 \in \mathcal{A}_a, \\ & \alpha_1 = (\text{sn}, \text{RelateTo}, st_{\text{sm}_1}, \text{sm}_1, h_1), \alpha_2 = (\text{sn}, \text{RelateTo}, st_{\text{sm}_2}, \text{sm}_2, h_2) \\ & \Rightarrow \alpha_1 = \alpha_2. \end{aligned}$$

- (3) There is no other annotation $a_1 \in A(k-1)$ that shares signs of annotation with a , hence:

$$\nexists a_1 \in A(k-1) \mid SN_a \cap SN_{a_1} \neq \emptyset.$$

- (4) If the annotation $a \in A(k)$ annotates or relates to another annotation $a_1 \in A(k-1)$, then scope and access permission conflicts have to be avoided. Let us define the *conflict detector* function, $\text{cd} : A(k) \times A(k-1) \rightarrow \{0, 1\}$, so that:

$$\text{cd}(a, a_1) = \begin{cases} 0 & \text{if there are neither scope conflicts nor access permission conflicts} \\ 1 & \text{if there are either scope conflicts or access permission conflicts.} \end{cases}$$

Therefore, the following condition must be satisfied:

$$\forall h \in H_a \mid \text{h}(h) = a_1 \in A(k-1) \Rightarrow \text{cd}(a, a_1) = 0.$$

In conclusion, the first part of the annotation tuple is devoted to providing information about the annotation itself, because it specifies the handle of the annotation, its author, its groups of users with their respective access permissions, its scope, the signs of the annotation, and the link types. On the other hand, the second part of the annotation tuple provides information about the annotated or related digital objects, specifying which segment of which stream of which digital object is being annotated or related to, as shown in the following (we do not use the time dimension notation for space reasons, as it is not needed for this observation):

$$a = \left(\underbrace{h_a, au_a, G_a \times P, sp_a, \mathcal{A}_a \subseteq SN \times LT}_{\text{information about the annotation}} \times \underbrace{ST \times SM \times H}_{\text{information about the digital object}} \right).$$

Note that the author, au_a , of the annotation is not taken from the set of authors but from the set of users at time $k-1$. Indeed, from Definition 5.2, an author is a user who authored at least one digital object; thus, if we had used the set of authors in the definition of annotation, we would have constrained the author of the annotation to have authored at least one other digital object besides the annotation in question. In contrast, if we pick out a user from the set of users at time $k-1$, we allow that user to become an author at time k , simply because he is authoring the annotation at hand. Moreover, in contrast

to the case of the generic digital object introduced in Section 5.1, the annotation is constrained to be authored by one, and only one, author.

In the following section, we will discuss the meaning of the \mathcal{A}_a relation and the four constraints introduced in the definition of annotation in more detail.

Discussion About the \mathcal{A}_a Relation

The \mathcal{A}_a relation makes extensive use of the mechanism introduced in Section 4.2 for addressing the time dimension. In particular, the \mathcal{A}_a relation aims to demonstrate that an annotation must annotate or relate to digital objects that already exist. For this reason, in Definition 9.1, the annotation a belongs to $A(k)$, while the annotated or related digital objects belong to $DO(k-1)$, and are identified by their handles in $H(k-1)$. This notation underlines the fact that the annotation belongs to the set of digital objects at time k , but it works with the previously existing digital objects that belong to the set of digital objects at time $k-1$. Therefore, an annotation can only annotate or relate to already existing digital objects, which is quite intuitive but needs to be properly formalized.

A very important consequence of this choice is that:

$$h_a \notin H_a.$$

In fact, $\{h_a\} = H(k) \setminus H(k-1)$ while $H_a \subseteq H(k-1)$: h_a is precisely the handle identified by the transition from $H(k-1)$ to $H(k)$, as explained in Section 4.2. Therefore, an annotation cannot be self-referential: it cannot annotate or relate to itself, since a self-referential annotation would be useless.

The \mathcal{A}_a relation makes use of the set of signs $SN(k)$ at time k to indicate that they represent the signs created precisely for the annotation a . Furthermore, \mathcal{A}_a uses the set of segments $ST(k)$ at time k to indicate that those segments are created solely to allow the annotation a to point to the requested part of the streams contained in $SM(k-1)$. If we consider the mechanism introduced in Section 4.2 for formalizing the temporal dimension, when at time $k-1$ we pick out a new segment $st_{sm} \in \overline{ST}(k-1) \subseteq U_{ST}$, it can refer to a stream $sm \in SM(k-1)$: in fact, that stream already exists at time $k-1$, even though the new segment belongs to the set of segment $ST(k)$ only at time k . Note that the \mathcal{A}_a relation uses the set of streams $SM(k-1)$ at time $k-1$ because those are the streams that belong to the digital objects identified by their handles in $H(k-1)$. In conclusion, we deal with digital objects and their corresponding streams which already exist at time $k-1$ and which are annotated or related by using signs and segments that have just been created for the annotation a at time k .

In the \mathcal{A}_a relation, both segments and streams play a very important role in allowing an annotation to annotate, or relate to, the requested part of a digital object. In this context, the distinction between logical and physical streams and the possibility of using the logical streams as representatives of their equivalence classes, becomes a fundamental issue. We can always suppose that an annotation deals with logical streams, because the mapping to different

physical streams is correctly managed by the notion of the stream mapping relation, as introduced in Section 6.2. In this way, an annotation can annotate a logical stream and it is also implicitly annotating all of the physical streams that are in the same equivalence class of the logical stream. Furthermore, as discussed in Section 6.2, an annotation could annotate abstract streams belonging to equivalence classes created on the quotient set SM/SM . In this way, it obtains access to an entire hierarchy of different representations of the content of a digital object. Last, logical streams simplify the use of segments because we can always refer to contiguous indices in the logical streams even though they are not contiguous in the physical streams, as observed in Section 6.3. This possibility makes it easier to determine which part of the digital object is being annotated or related, because we can always make the assumption that we are dealing with contiguous indices in the stream of the digital object.

Last, the \mathcal{A}_a relation does not explicitly make use of the meanings of the annotation, even though they are a fundamental part of our model. As explained in Section 8, the meanings of annotation represent a kind of preexisting and superimposed knowledge which does not belong to any specific sign of annotation in particular, but rather should be shared by different signs of annotation to support cooperation and interoperability. In this sense, the meanings of annotation are not directly part of any specific annotation; on the other hand, as introduced in Definition 8.1, the meanings function, m , allows us to associate each sign of annotation with its corresponding meaning of annotation. Therefore, for each sign, $sn \in SN_a$, we can use $m(sn)$ to obtain its meanings of annotation, and we can then exploit and navigate the meanings graph, if necessary.

Discussion about the Constraints of the Annotation. The first two constraints are *intra-annotation* constraints, because they limit the \mathcal{A}_a relation, which is the core of the annotation; the second two constraints instead are *inter-annotation* constraints, because they regulate the relationships of the annotation with respect to other annotations.

The first constraint imposes the existence and uniqueness of the annotated digital object and prevents the annotated digital object from being related as well. In this way, the constraint introduced in Section 6.1:

an annotation must annotate one and only one digital object, either a document or another annotation, hence an annotation must have one and only one “annotate link,”

is complied with. Furthermore, it enforces the distinction between the “annotate link” and the “relate-to link,” because the annotated digital object cannot also be related. Therefore, it underlines the fact that the role of the “annotate link” is to express intradigital object relationships, while the “relate-to link” is requested to express only interdigital object relationships. Furthermore, each sign must cooperate—once and only once—in expressing such intra-DO relationships: there is no sign whose only link is the “relate-to link.” A consequence

of this constraint is:

$$\begin{aligned} \forall h_1, h_2 \in H_a \mid \exists \alpha_1, \alpha_2 \in \mathcal{A}_a, \\ \alpha_1 = (\text{sn}_1, \text{Annotate}, st_{\text{sm}_1}, \text{sm}_1, h_1), \alpha_2 = (\text{sn}_2, \text{Annotate}, st_{\text{sm}_2}, \text{sm}_2, h_2) \\ \Rightarrow h_1 = h_2. \end{aligned}$$

The second constraint aims at keeping the semantics of a sign as clear as possible: if a sign sn could be related to more digital objects, it would not be clear which of its meanings—given by $m(\text{sn})$ —should be applied to each related digital object. In conclusion, this constraint together with the first constraint states that a sign of annotation must annotate one and only one segment of a digital object and it may relate to one and only one segment of another digital object.

The third constraint ensures that the signs of an annotation are not shared with any other annotation to preserve the mechanism of sharing common semantics among annotations. As explained in Section 8, the sharing of meanings of annotation by means of the \mathcal{M}_1 and \mathcal{M}_2 relations is the mechanism for pointing out common semantics among annotations; on the other hand, the direct sharing of signs of annotation could be misleading. In fact, a sign is a materialization of a meaning: the same sign might be used by different users with completely different semantics while different signs used by different users might have the same semantics. Consider, for example, two users who use the star symbol: one uses it to indicate an important passage, while the other uses it to indicate a wrong passage; in this case, we have two signs that look exactly the same but have two completely different meanings. Another situation would be where two users use the star symbol and the exclamation mark, both to indicate an important passage; in this case, we have two signs that look different but have the same meaning. Therefore, the cooperation among users happens by sharing common meanings, which are connected by way of the meanings graph, and not by directly sharing signs, which may be misleading or incoherent. Finally, note that this constraint does not prevent the existence of two signs looking exactly the same, but it means that these two signs are different elements in the set SN .

The fourth constraint prevents pathologic situations such as, for example, a public annotation that annotates or relates to a private annotation. In such cases there is a *scope conflict*: using the previously mentioned example, the author of the private annotation can see both the public and the private annotations, but another user can see only the public annotation, which is annotating something hidden to this user. As a further example, a shared annotation, for which a given group has “denied” access permission, should not be annotated by, or related to, another shared annotation, for which the same group has “read and write” access permission, because in this case we would obtain an *access permission conflict*. The situation may be even more complicated if both scope and access permission conflicts happen at the same time.

To avoid such situations, the *conflict detector* cd function has been introduced, which returns 1 if there is any kind of conflict, and 0 otherwise. As we introduced in Sections 5.2 and 5.3, the concepts of access permission and scope

of annotation are not intended to enforce a specific access policy, but rather to provide us with the means of expressing any necessary access policy. As a consequence, the actual definition of the conflict detector function has to be done case by case in order to carry out the necessary access policy: we cannot provide an a priori definition of it that is appropriate for all cases.

For example, if we set the following rules:

- A public annotation, a_1 , can be freely annotated or related to by any annotation a , without further restrictions.
- A shared annotation, a_1 , can be annotated or related to only by a shared or private annotation a . In the case of a private annotation a , the author of the annotation a must belong to at least one of the groups of users sharing the annotation a_1 , provided that the access permission for that group is not denied in a_1 . In the case of a shared annotation a , all of the groups of users sharing the annotation a must also be sharing the annotation a_1 , provided that none of them has been denied access permission in a_1 . It follows that a shared annotation cannot be annotated by or related to a public annotation;
- A private annotation a_1 can be annotated or related to only by a private annotation a , provided that they have the same author. It follows that a private annotation can be annotated by, or related to, neither a public annotation nor a shared annotation;

then we can provide the following definition of the conflict detector function:

$$cd(a, a_1) = \begin{cases} 0 & \text{if } sp_{a_1} = \text{Public} \\ 0 & \text{if } sp_{a_1} = \text{Shared} \wedge sp_a = \text{Private} \wedge \exists (G_1, p_1) \in G_{a_1} \times P \mid \\ & au_a \in G_1 \wedge p_1 \neq \text{Denied} \\ 0 & \text{if } sp_{a_1} = sp_a = \text{Shared} \wedge \forall (G, p) \in G_a \exists (G_1, p_1) \in G_{a_1} \mid \\ & G = G_1 \wedge p_1 \neq \text{Denied} \\ 0 & \text{if } sp_{a_1} = sp_a = \text{Private} \wedge au_a = au_{a_1} \\ 1 & \text{otherwise.} \end{cases}$$

10. DOCUMENT-ANNOTATION HYPERTEXT

As explained in Section 3, we consider that existing digital objects and annotations constitute a hypertext. The definition and the properties of this hypertext directly follow from the definition of annotation we provided in the previous sections. Therefore, we can consider the document-annotation hypertext as a kind of view of the set of documents and annotations. The aim is to mask all of the details involved by the definition of the annotation itself, and to provide us with a more abstract representation of the objects we dealt with and of their structural relationships.

We will introduce the definition of document-annotation hypertext and we will study its properties by directly using the set of digital objects, DO , and the set of annotations, A . In the following, we will not make use of the set of handles, H , as might be expected from the previous discussion where annotations can be linked to digital objects only by using their handles. This choice allows us to explain the properties of the document-annotation hypertext in a clearer and more intuitive way than doing so by using handles, which would

just add a further level of indirection. Indeed, if we used handles instead of digital objects in the explanation, we would have to map each handle back to the corresponding digital object by using the h function, in order to exploit the characteristics of the digital object in the reasoning about the document-annotation hypertext. On the other hand, since the h function is bijective, we are sure that the properties of the document-annotation hypertext demonstrated by directly using digital objects hold, and are valid even in the case of the use of handles. This is quite important because according to the line of reasoning developed in the previous sections we may not directly deal with documents, which could be independently managed by external DLMS, but we will always have the possibility of referring to those digital objects by using their handles. In conclusion, the actual document-annotation hypertext could be constructed by using the handles of the digital objects, even though its properties are better explained and demonstrated by directly using the digital objects.

Definition 10.1. The *document-annotation hypertext* is a labeled directed graph:

$$(H_{da} = (DO, E_{da} \subseteq A \times DO), l_{da}),$$

where:

- $DO = A \cup D$ is a set of vertices;
- $E_{da} = \{(a, do) \in A \times DO \mid \exists \alpha \in \mathcal{A}_a, \alpha = (sn, t, st_{sm}, sm, h^{-1}(do))\}$ is a set of edges;
- $l_{da} : E_{da} \rightarrow LT$ is a labelling function, such that for each $e = (a, do) \in E_{da}$ there is a lt -labeled edge from the annotation a to the generic digital object do :

$$l_{da}(a, do) = \begin{cases} \text{Annotate} & \text{if } \exists \alpha \in \mathcal{A}_a \mid \alpha = (sn, \text{Annotate}, st_{sm}, sm, h^{-1}(do)) \\ \text{RelateTo} & \text{if } \exists \alpha \in \mathcal{A}_a \mid \alpha = (sn, \text{RelateTo}, st_{sm}, sm, h^{-1}(do)). \end{cases}$$

The document-annotation hypertext is constructed by putting an edge between an annotation vertex and a digital object vertex, if the annotation is annotating or relating to that digital object. Note that we used $h^{-1}(do)$ in E_{da} to track the digital object back to its handle; the edge is then labeled with the corresponding link type. Each edge $e = (a, do) \in E_{da}$ always starts from an annotation $a \in A$, while $e \in E_{da}$, which starts from a document $d \in D$, does not exist.



Note that we deal with a graph H_{da} and not with a multigraph—a graph where multiple edges between the same vertices are allowed—as may happen in the case of an annotation relating to a different part of the same digital object. Therefore, we consider that multiple edges with the same direction between the same vertices are collapsed into a single edge.

Table II summarizes the graphical conventions, adopted in the following figures.

Figure 3 shows an example of document-annotation hypertext H_{da} :

- $D = \{d_1, d_2, d_3, d_4, d_5\}$; we can assume that the subscript of each document indicates the time in which the document became an element of the set D .

Table II. Graphical Conventions

	Annotate Link	RelateTo Link
Document 	continuous line labeled A	dotted line labeled R
Annotation 	continuous line labeled A	dotted line labeled R

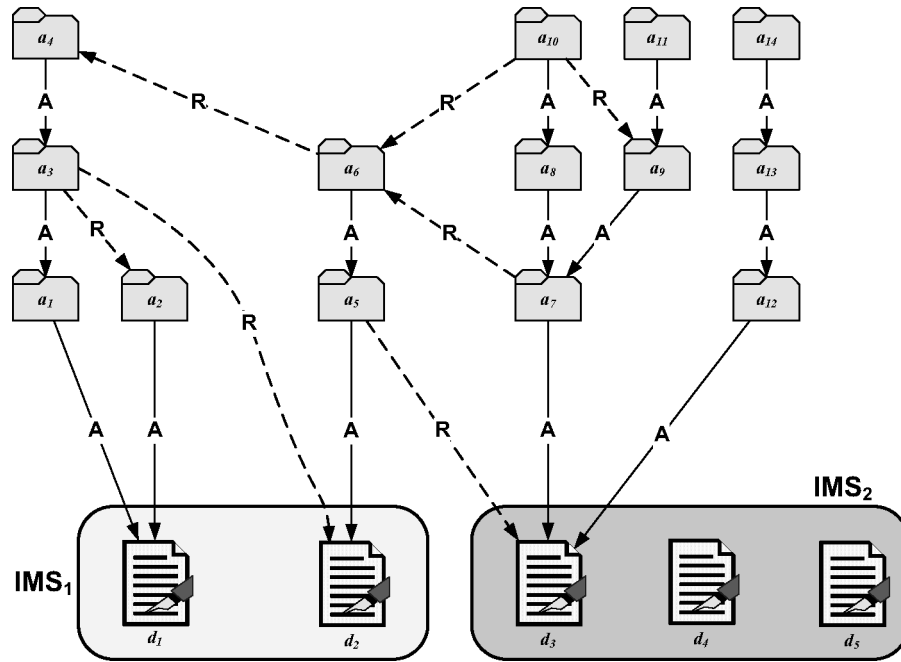


Fig. 3. Example of document-annotation hypertext H_{da} .

- $A = \{a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, a_{14}\}$; we can assume that the subscript of each annotation indicates the time in which the annotation became an element of the set A .
- We can express, for example:
 - *Annotation sets concerning a document*: $\{a_1, a_2\}$ is an annotation set concerning the document d_1 .
 - *Annotation sets concerning an annotation*: $\{a_8, a_9\}$ is an annotation set concerning the annotation a_7 .
 - *Annotation threads concerning a document*: $\{a_1, a_3, a_4\}$ is an annotation thread concerning the document d_1 .
 - *Annotation threads concerning an annotation*. $\{a_8, a_{10}\}$ is an annotation thread concerning the annotation a_7 ;
 - *Multiple annotation threads concerning a document*: $\{a_7, a_8, a_{10}\}$ and $\{a_{12}, a_{13}, a_{14}\}$ are two different annotations threads, both concerning the document d_3 .

- Multiple annotation threads concerning an annotation:* $\{a_8, a_{10}\}$ and $\{a_9, a_{11}\}$ are two annotation threads both concerning the annotation a_7 .
- Nested annotation threads concerning a document:* $\{a_8, a_{10}\}$ and $\{a_9, a_{11}\}$ are two different and nested annotation threads both concerning the document d_3 .

Figure 3 also points out another important feature of the document-annotation hypertext. It can span and cross the boundaries of the single IMS, as discussed in Section 3. IMS_1 manages d_1 and d_2 , while IMS_2 manages d_3 , d_4 , and d_5 . There are annotations that act as a bridge between two IMSs: for example, a_5 annotates d_2 , which is managed by IMS_1 , and refers to d_3 , which is managed by IMS_2 .

PROPOSITION 10.2. *The document-annotation hypertext has the following properties:*

- (1) *The graph does not contain loops:*

$$\forall a \in A, \nexists e = (a, do) \in E_{da} \mid a = do.$$

- (2) *Each annotation a is incident with one and only one edge labeled Annotate.*

$$\forall a \in A, \exists! e = (a, do) \in E_{da} \mid l_{da}(e) = \text{Annotate}$$

- (3) *The graph does not contain cycles:*

$$\nexists C = a_1 a_k a_{k-1} \cdots a_2 a_1 \mid$$

$$e_1 = (a_1, a_k), e_k = (a_k, a_{k-1}), \dots, e_2 = (a_2, a_1) \in E_{da}, \quad k > 1.$$

- (4) *Given a set $A' \subset A$ there are at least $|A'|$ edges in H_{da} incident on elements of A' . Therefore, the following relationship holds for the size of H_{da} :*

$$\varepsilon(H_{da}) \geq |A|.$$

PROOF. We can show that:

- (1) From Definition 9.1, it follows that $h_a \notin H_a$ and, as explained in Section 9, $\nexists \alpha \in \mathcal{A}_a \mid \alpha = (\text{sn}, t, st_{\text{sm}}, \text{sm}, h_a)$; thus, $\nexists e = (a, a) \in E_{da}$.
- (2) From Definition 9.1, we have the following constraint $\exists! h \in H_a \mid \forall \text{sn} \in SN_a, \exists! \alpha \in \mathcal{A}_a, \alpha = (\text{sn}, \text{Annotate}, st_{\text{sm}}, \text{sm}, h)$; it follows that $\forall a \in A, \exists! do \in DO \mid \exists \alpha \in \mathcal{A}_a, \alpha = (\text{sn}, \text{Annotate}, st_{\text{sm}}, \text{sm}, h^{-1}(do))$; thus, there exists a unique edge such that $l_{da}(a, do) = \text{Annotate}$.
- (3) Annotations entail a temporal dimension, since each annotation must annotate or relate to an already existing digital object, as explained in Sections 4.2 and 9. From Definition 9.1, the \mathcal{A}_a relation involves the set $H(k-1)$, of handles of digital objects that already belong to the set of digital objects at time $k-1$, while the annotation belongs to the set $A(k)$. Therefore, by means of the \mathcal{A}_a relation an annotation $a \in A(k)$ can annotate or relate to only digital objects that already exist at time k , an annotation cannot annotate or relate to another annotation that does not already exist at time k . It follows cycles, such as the one shown in Figure 4, where the oldest annotation $a_1 \in A(1)$ annotates or relates to the newest annotation

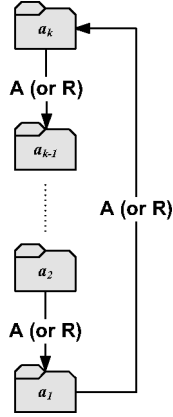


Fig. 4. Disallowed annotations cycle.

$a_k \in A(k)$ with $k > 1$, are not possible. In fact, when the oldest annotation $a_1 \in A(1)$ was created at time 1, the newest annotation $a_k \in A(k)$, $a_k \notin A(0)$ did not exist and so it could not have been involved in \mathcal{A}_{a_1} , which makes use of digital objects belonging to the set of digital objects at time 0. Note that this issue does not exist for document $d \in D$ vertices, since edges can start only from annotation vertices.

- (4) Since for item number 2, each annotation a must be incident with one and only one Annotate edge, then for $|A'|$ annotations there are, at least $|A'|$ edges; there may be more if there are also RelateTo edges. In H_{da} there are $|A|$ annotations, and therefore $\varepsilon(H_{da}) \geq |A|$. \square

Proposition 10.2 expresses the constraints imposed on the annotation in Definition 9.1 in terms of a graph: first, the graph does not contain loops corresponding to self-referential annotations that are useless for our purposes; second, each annotation is incident with one and only one edge of the kind “Annotate link,” thus formalizing the constraint on the link types introduced in Section 6.1; third, since each annotation can annotate or relate to an already existing DO, the third property ensures that there are no cycles where the oldest annotation a_1 annotates or relates to the newest annotation a_k , as shown in Figure 4; last, the fourth property sets a lower bound to the size of H_{da} .

Figure 5 shows the patterns that can be obtained by combining the allowed link types: note that each pattern is characterized by only one edge of the type “Annotate link”; furthermore an annotation is not allowed to exclusively have “RelateTo link” edges.

Note that the example of document-annotation hypertext shown in Figure 3 is the result of the combination of these basic allowed annotation patterns.

PROPOSITION 10.3. *Let $H'_{da} = (DO', E'_{da})$ be the subgraph of H_{da} , such that:*

- $E'_{sda} = \{e \in E_{da} \mid l_{da}(e) = \text{Annotate}\}$.
- $DO' = \{do \in DO \mid \exists e' \in E'_{da}, e' = (a, do)\}$.

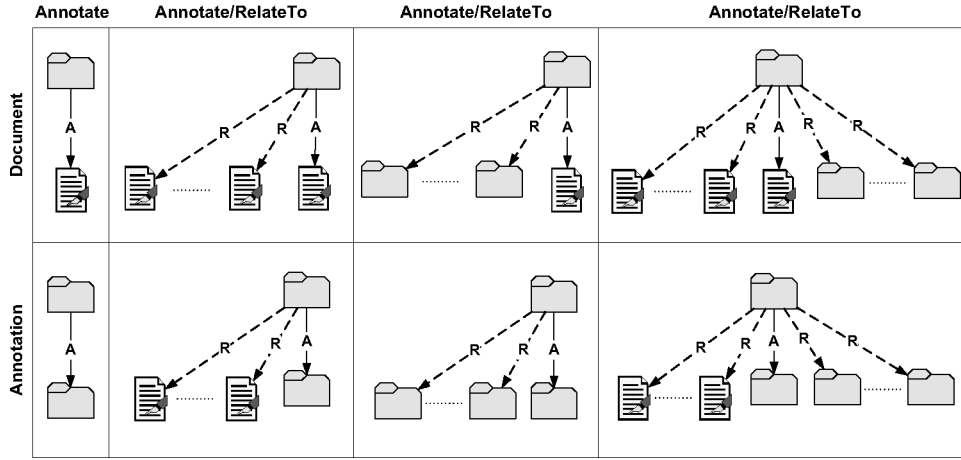


Fig. 5. Allowed annotation patterns.

H'_{da} is the subgraph whose edges are of the kind, Annotate, and whose vertices are incident with at least one of these edges. Let $H''_{da} = (DO'', E''_{da})$ be the underlying graph of H'_{da} , which is the undirected version of H'_{da} .

The following properties hold:

- (1) $\varepsilon(H'_{da}) = \varepsilon(H''_{da}) = |A|$.
- (2) H''_{sda} is a forest.
- (3) Every tree in H''_{da} contains a unique document vertex d .

PROOF. We can show that:

- (1) According to Proposition 10.2, $\varepsilon(H'_{da}) = \varepsilon(H''_{da}) \geq |A|$ but since in H'_{da} and H''_{da} , there are only Annotate edges, we have $\varepsilon(H'_{da}) = \varepsilon(H''_{da}) = |A|$.
- (2) Ab absurdo: if H''_{da} was not a forest, then it would be a cyclic graph. The only way of obtaining a cycle in H''_{da} is that in H_{da} :

$$\begin{aligned} \exists a \in A, \exists e_1 = (a, do_1), e_2 = (a, do_2) \in E_{da}, do_1 \neq do_2 \mid \\ l_{da}(e_1) = l_{da}(e_2) = \text{Annotate}, \end{aligned}$$

an annotation exists in H_{da} from which two edges of the kind, Annotate, start, but this contradicts the Definition 10.1 given for the graph H_{da} and thus, H''_{da} is a forest;

- (3) Since H''_{da} is a forest, its components are trees. Ab absurdo, suppose that there is a tree T whose vertices are only annotations. A tree T with n vertices has $n - 1$ edges but, for proposition 10.2, in H_{da} (and also in H''_{da}), n annotations are incident with n edges; so T cannot be a tree. Therefore, every tree in H''_{da} contains at least one document vertex d .

Suppose now that there is a tree T , which contains two document vertices, d_1 and d_2 , $d_1 \neq d_2$. Since for every two vertices in a tree there is a unique path connecting them, in the path $P = d_1 a_1 \dots a_i \dots a_k d_2$ there must be

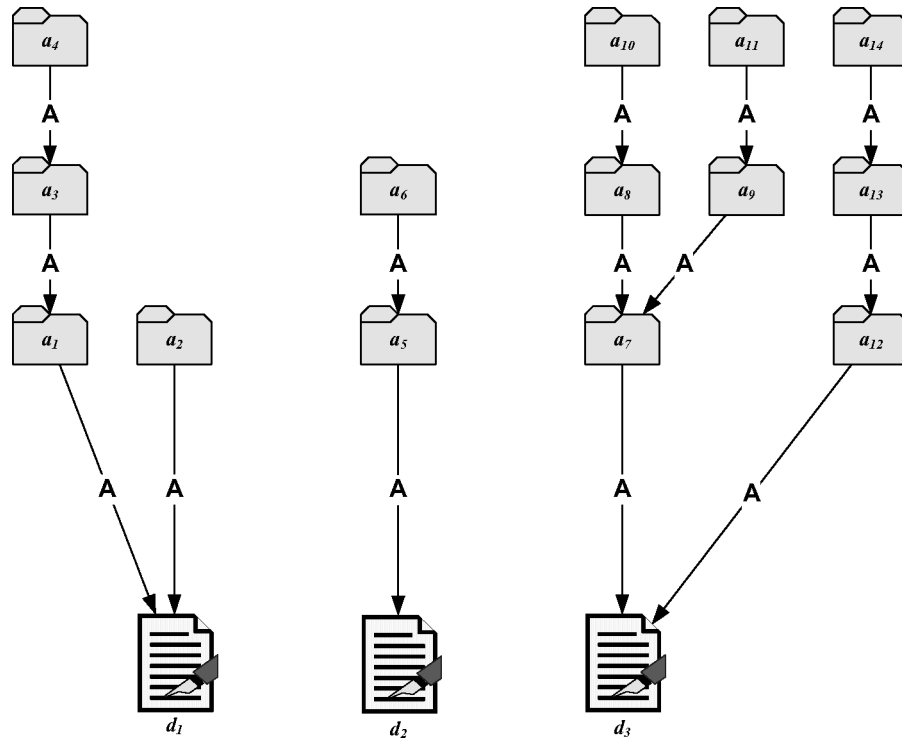


Fig. 6. Example of the H'_{da} subgraph, obtained from the document-annotation hypertext H_{da} of Figure 3.

an annotation a_i from which in H_{da} , two edges of the kind Annotate start; since by definition of H_{da} there are no edges of the type $e = (d_m, d_n) \in E_{da}$. However, the annotation a_i contradicts the definition of H_{da} , and thus there is a unique document vertex d in T . \square

Note that if we had not removed the “RelateTo link” edges from the graph H'_{da} , it could have contained cycles. Consider Figure 3: for example, a cycle would be $C = a_7a_6a_{10}a_8a_7$, because in H'_{da} we do not consider the direction of the edges.

Figure 6 shows an example of the H'_{da} subgraph, obtained from the document-annotation hypertext H_{da} of Figure 3.

11. CONCLUSIONS AND FUTURE WORK

We have discussed the problem of providing users with annotations on different kinds of digital objects managed by IMS that can range from DBMS to DLMS. In particular, we have addressed this problem in the context of DLMS and their evolution [DELOS 2004; Ioannidis et al. 2005; Candela et al. 2006]. To this end, annotations have been studied and formalized as an effective tool suitable for enabling and carrying out the evolution of DLMS.

We have introduced the distinction between the meaning and the sign of an annotation. This distinction has allowed us to better describe both the semantics and the materialization of the annotation and to adopt a flexible approach in modeling annotations. In fact, this gives us the opportunity to deal with the semantics of annotations in a flexible way, avoiding predefined types, and making it possible to exploit them as an effective collaboration tool for users.

We have proposed a formal model of annotations on digital content, which until now has been absent from the literature concerning annotations. This formal model not only captures all the aspects we have described, but it also effectively formalizes the time dimension entailed by annotations. Furthermore, it introduces the notion of document-annotation hypertext and explores some of its properties. Finally, it provides us with a sound theoretical basis for future research on this matter.

Future research work will concern the use of annotations in order to search for documents, and the proposed formal model constitutes the necessary groundwork to be able to design and formalize search algorithms and to express query languages that take annotations into account. Annotations provide us with an additional source of evidence, which is complementary to that already contained in the set of documents. Therefore, we can exploit annotations with the two final goals of retrieving more relevant documents and ranking them better. Furthermore, the paths that connect annotations to documents become the vehicle for moving this further source of evidence towards the documents. In fact, the document-annotation hypertext is the basic infrastructure for combining the sources of evidence that come from documents and annotations. We have already started to work on this problem in the context of *data fusion* [Agosti and Ferro 2005b, 2006]. This is because we need to combine the source of evidence, which comes from annotations with the source which comes from documents. For the future, we plan to employ both hypertext information retrieval [Agosti and Smeaton 1996] and *link fusion* techniques [Xi et al. 2004] for designing advanced search algorithms that involve annotations based on our formal model.

Once we have developed search strategies that exploit annotations, we will therefore need to evaluate their retrieval performances by using standard information retrieval methodologies. We plan to adopt the Cranfield methodology [Cleverdon 1997], which makes use of experimental collections to measure the performances of an information retrieval system. The performances are measured by using the standard precision and recall figures [van Rijsbergen 1979; Salton and McGill 1983], but according to Hull [1993] we also need a statistical methodology for judging whether the measured performances can be considered statistically significant. The next step will be to investigate the possibility of using measures that differ from precision and recall and are better tailored to the features of annotations. Finally, there is a lack of experimental test collections with annotated digital contents. We have already started to work on this problem [Agosti et al. 2007b] and our future research work will also concern the design and development of this kind of test collection.

ACKNOWLEDGMENTS

Sincere thanks are due to Massimo Melucci, Nicola Orio, and Luca Pretto for the time they spent with the authors discussing aspects related to annotations and their relationship with hypertexts. The authors would like to warmly thank Costantino Thanos, coordinator of the DELOS network of excellence on digital libraries, for his continuous support and advice. We thank the anonymous referees for their fruitful comments and careful reading, which improved this article.

REFERENCES

- ABITEBOUL, S., AGRAWAL, R., BERNSTEIN, P., CAREY, M., CERI, S., CROFT, B., DEWITT, D., FRANKLIN, M., GARCIA-MOLINA, H., GAWLICK, D., GRAY, J., HAAS, L., HALEVY, A., HELLERSTEIN, J., IOANNIDIS, Y., KERSTEN, M., PAZZANI, M., LESK, M., MAIER, D., NAUGHTON, J., SCHEK, H.-J., SELLIS, T., SILBERSCHATZ, A., STONEBRAKER, M., SNODGRASS, R., ULLMAN, J. D., WEIKUM, G., WIDOM, J., AND ZDONIK, S. 2005. The Lowell Database research self-assessment. *Comm. ACM* 48, 5, 111–118.
- AGOSTI, M., ALBRECHTSEN, H., FERRO, N., FROMMHOLZ, I., HANSEN, P., ORIO, N., PANIZZI, E., PEJTERSEN, A. M., AND THIEL, U. 2005c. DiLAS: a Digital library annotation service. In *Proceedings of the International Workshop on Annotation for Collaboration – Methods, Tools, and Practices (IWAC 2005)*, J.-F. Boujut, Ed. CNRS—Programme société de l’information, 91–101.
- AGOSTI, M., BENFANTE, L., AND ORIO, N. 2003. IPSA: a digital archive of herbals to support scientific research. In *Proceedings of the 6th International Conference on Asian Digital Libraries—Digital Libraries: Technology and Management of Indigenous Knowledge (ICADL’03)*, T. M. T. Sembok, H. B. Zaman, H. Chen, S. R. Urs, and S. H. Myaeng, Eds. Lecture Notes in Computer Science, vol. 2911, Springer, Heidelberg, Germany, 253–264.
- AGOSTI, M., BONFIGLIO-DOSIO, G., AND FERRO, N. 2007a. A historical and contemporary study on annotations to derive key features for systems design. *Int. J. Dig. Lib.* <http://dx.doi.org/10.1007/s00799-007-0010-0>.
- AGOSTI, M., CANDELA, L., CASTELLI, D., FERRO, N., IOANNIDIS, Y., KOUTRIKA, G., MEGHINI, C., PAGANO, P., ROSS, S., SCHEK, H.-J., AND SCHULDT, H. 2006a. A Reference Model for DLMSs Interim report. In *Deliverable D1.4.2 - Reference Model for Digital Library Management Systems [Draft 1]*, L. Candela and D. Castelli, Eds. DELOS, A Network of Excellence on Digital Libraries—IST-2002-2.3.1.12, Technology-Enhanced Learning and Access to Cultural Heritage. http://146.48.87.122:8003/OLP/Repository/1.0/Disseminate/delos/2006_WP1_D142/content/pdf?version=1 [last visited 2007, March 23].
- AGOSTI, M., COLOTTI, R., AND GRADENIGO, G. 1991. A two-level hypertext retrieval model for legal data. In *Proceedings of the 14th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR’91)*, E. A. Fox, Ed. ACM Press, New York, 316–325.
- AGOSTI, M., COPPOTELLI, T., FERRO, N., AND PRETTO, L. 2007b. Exploiting relevance assessment for the creation of an experimental test collection to evaluate systems that use annotations. In *DELOS Conference 2007 Working Notes*, C. Thanos and F. Borri, Eds. ISTI-CNR, Gruppo ALI, Pisa, Italy, 195–202.
- AGOSTI, M. AND FERRO, N. 2003a. Annotations: Enriching a digital library. In *Proceedings of the 7th European Conference on Research and Advanced Technology for Digital Libraries (ECDL’03)*, T. Koch and I. T. Sølberg, Eds. Lecture Notes in Computer Science, vol. 2769, Springer, Heidelberg, Germany, 88–100.
- AGOSTI, M. AND FERRO, N. 2003b. Chapter X: Managing the interactions between handheld devices, mobile applications, and users. In *Advances in Mobile Commerce Technologies*, E. P. Lim and K. Siau, Eds. Idea Group, Hershey, PA, 204–233.
- AGOSTI, M. AND FERRO, N. 2004. An information service architecture for annotations. In *Digital Library Architectures: Peer-to-Peer, Grid, and Service-Oriented, Pre-proceedings of the 6th Thematic Workshop of the EU Network of Excellence DELOS*, M. Agosti, H.-J. Schek, and C. Türker, Eds. Edizioni Libreria Progetto, Padova, Italy, 115–126.
- AGOSTI, M. AND FERRO, N. 2005a. A system architecture as a support to a flexible annotation service. In *Peer-to-Peer, Grid, and Service-Oriented in Digital Library Architectures: 6th Thematic*

- Workshop of the EU Network of Excellence DELOS. Revised Selected Papers*, C. Türker, M. Agosti, and H.-J. Schek, Eds. Lecture Notes in Computer Science, vol. 3664, Springer, Heidelberg, Germany, 147–166.
- AGOSTI, M. AND FERRO, N. 2005b. Annotations as context for searching documents. In *Proceedings of the 5th International Conference on Conceptions of Library and Information Science—Context: Nature, Impact and Role*, F. Crestani and I. Ruthven, Eds. Lecture Notes in Computer Science, vol. 3507, Springer, Heidelberg, Germany, 155–170.
- AGOSTI, M. AND FERRO, N. 2006. Search strategies for finding annotations and annotated documents: the FAST Service. In *Proceedings of the 7th International Conference on Flexible Query Answering Systems (FQAS'06)*, H. Legind Larsen, G. Pasi, D. Ortiz-Arroyo, T. Andreassen, and H. Christiansen, Eds. Lecture Notes in Artificial Intelligence 4027, Springer, Heidelberg, Germany, 270–281.
- AGOSTI, M., FERRO, N., FROMMHOLZ, I., PANIZZI, E., PUTZ, W., AND THIEL, U. 2006b. Integration of the DiLAS annotation service into digital library infrastructures. In *Proceedings of the 1st International Workshop on Digital Libraries in the Context of Users' Broader Activities (DL-CUBA'06)*, A. Blandford and J. Gow, Eds. <http://www.ucl.ac.uk/events/dl-cuba2006/> [last visited 2007, March 23], 1–4.
- AGOSTI, M., FERRO, N., FROMMHOLZ, I., AND THIEL, U. 2004. Annotations in digital libraries and laboratories—facets, models and usage. In *Proceedings of the 8th European Conference on Research and Advanced Technology for Digital Libraries (ECDL'04)*, R. Heery and L. Lyon, Eds. Lecture Notes in Computer Science, vol. 3232, Springer, Heidelberg, Germany, 244–255.
- AGOSTI, M., FERRO, N., AND ORIO, N. 2005a. Annotating illuminated manuscripts: An effective tool for research and education. In *Proceedings of the 5th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL'05)*, M. Marilino, T. Sumner, and F. Shipman, Eds. ACM Press, New York, 121–130.
- AGOSTI, M., FERRO, N., AND ORIO, N. 2005b. Annotations as a support to research users. In *Proceedings of the 7th International Workshop of the EU Network of Excellence DELOS on Audio-Visual Content and Information Visualization in Digital Libraries (AVIVDiLib'05)*, T. Catarci, S. Christodoulakis, and A. Del Bimbo, Eds. Centromedia, Viareggio, Italy, 117–120.
- AGOSTI, M., FERRO, N., AND ORIO, N. 2006c. Graph-based automatic suggestion of relationships among images of illuminated manuscripts. In *Proceedings of the 21st ACM Symposium on Applied Computing (SAC'06)*, H. M. Haddad, K. M. Liebrock, R. Chbeir, M. J. Palakal, S. Ossowski, K. Yetongnoon, R. L. Wainwright, and C. Nicolle, Eds. ACM Press, New York, 1063–1067.
- AGOSTI, M., FERRO, N., PANIZZI, E., AND TRINCHESE, R. 2006d. Annotation as a support to user interaction for content enhancement in digital libraries. In *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI'06)*, A. Celentano and P. Mussio, Eds. ACM Press, New York, 151–154.
- AGOSTI, M. AND SMEATON, A., EDS. 1996. *Information Retrieval and Hypertext*. Kluwer Academic Publishers, Norwell, MA.
- BERNERS-LEE, T. 1994a. Uniform Resource Locators (URL). RFC 1738.
- BERNERS-LEE, T. 1994b. Universal Resource Identifiers in WWW. RFC 1630.
- BERNERS-LEE, T., FIELDING, R., IRVINE, U. C., AND MASINTER, L. 1998. Uniform Resource Identifiers (URI): Generic Syntax. RFC 2396.
- BERNERS-LEE, T., HENDLER, J., AND LASSILA, O. 2001. The semantic web. *Scientific American*, http://www.sciamdigital.com/index.cfm?fa=Products.ViewIssuePreview&ARTICLEID_CHAR=5704DA3F-F381-40DD-ABB2-52ECO2A748 [last visited 2007, March 23].
- BHAGWAT, D., CHITICARIU, L., TAN, W.-C., AND VIJAYVARGIYA, G. 2004. An Annotation Management System for Relational Databases. In *Proceedings of the 30th International Conference on Very Large Data Bases (VLDB'04)*, M. A. Nascimento, M. T. Özsu, D. Kossmann, R. J. Miller, J. A. Blakeley, and K. B. Schiefer, Eds. Morgan Kaufmann, 900–911.
- BOLLOBÁS, B. 1998. *Modern Graph Theory*. Springer-Verlag, New York.
- BOTTONI, P., CIVICA, R., LEVIALDI, S., ORSO, L., PANIZZI, E., AND TRINCHESE, R. 2004. MAD-COW: a Multimedia Digital Annotation System. In *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI'04)*, M. F. Costabile, Ed. ACM Press, New York, 55–62.

- BOTTONI, P., CIVICA, R., LEVIALDI, S., ORSO, L., PANIZZI, E., AND TRINCHESE, R. 2005a. Digital library content annotation with the MADCOW system. In *Proceedings of the 7th International Workshop of the EU Network of Excellence DELOS on Audio-Visual Content and Information Visualization in Digital Libraries (AVIVDiLib'05)*, T. Catarci, S. Christodoulakis, and A. Del Bimbo, Eds. Centromedia, Viareggio, Italy, 111–116.
- BOTTONI, P., CIVICA, R., LEVIALDI, S., ORSO, L., PANIZZI, E., AND TRINCHESE, R. 2005b. Storing and retrieving multimedia Web notes. In *Proceedings of the 4th International Workshop on Databases in Networked Information Systems (DNIS'05)*, S. Bhalla, Ed. Lecture Notes in Computer Science 3433, Springer, Heidelberg, Germany, 119–137.
- BOTTONI, P., COSTABILE, M. F., AND MUSSIO, P. 1999. Specification and dialogue control of visual interaction through visual rewriting systems. *ACM Trans. Prog. Lang. Syst.* 21, 6 (Nov.), 1077–1136.
- BOTTONI, P., LEVIALDI, S., AND RIZZO, P. 2003. An analysis and case study of digital annotation. In *Proceedings of the 3rd International Workshop on Databases in Networked Information Systems (DNIS '03)*, N. Bianchi-Berthouze, Ed. Lecture Notes in Computer Science 2822, Springer, Heidelberg, Germany, 216–230.
- BRUSH, A. J. B., BARGERON, D., GRUDIN, J., BORNING, A., AND GUPTA, A. 2002. Supporting interaction outside of class: Anchored discussion vs. discussion boards. In *Proceedings of the International Conference on Computer Support for Collaborative Learning (CSCL'02)*, G. Stahl, Ed. Lawrence Erlbaum Associates Inc., Mahwah, NJ, 425–434.
- BRUSH, A. J. B., BARGERON, D., GUPTA, A., AND CADIZ, J. J. 2001. Robust Annotation Positioning in Digital Documents. In *Proceedings of the Conference on Human Factors and Computing Systems (CHI'01)*, J. Jacko and A. Sears, Eds. ACM Press, New York, 285–292.
- BUNEMAN, P., KHANNA, S., TAJIMA, K., AND TAN, W.-C. 2004. Archiving scientific data. *ACM Trans. Data. Syst.* 29, 1 (Mar.), 2–42.
- BUNEMAN, P., KHANNA, S., AND TAN, W.-C. 2001. Why and where: A characterization of data provenance. In *Proceedings of the 8th International Conference on Database Theory (ICDT'01)*, J. Van den Bussche and V. Vianu, Eds. Lecture Notes in Computer Science 1973, Springer, Heidelberg, Germany, 316–330.
- BUNEMAN, P., KHANNA, S., AND TAN, W.-C. 2002. On propagation of deletions and annotations through views. In *Proceedings of the 21st ACM SIGMOD–SIGACT–SIGART Symposium on Principles of Database Systems (PODS'02)*, S. Abiteboul, P. G. Kolaitis, and L. Popa, Eds. ACM Press, New York, 150–158.
- CANDELA, L., CASTELLI, D., IOANNIDIS, Y., KOUTRIKA, G., PAGANO, P., ROSS, S., SCHEK, H.-J., AND SCHULDIT, H. 2006. The digital library manifesto. In *DELOS, A Network of Excellence on Digital Libraries—IST-2002-2.3.1.12, Technology-Enhanced Learning and Access to Cultural Heritage*. http://146.48.87.122:8003/OLP/Repository/1.0/Disseminate/delos/2006_other_0081/content/pdf?version=1 [last visited 2007, March 23].
- CANOVA, M. G. 1988. La tradizione europea degli erbari miniati e la scuola veneta. In *Di Sana Pianta. Erbari e Taccuini di Sanità: le Radici Storiche della Nuova Farmacologia*. Panini, Modena, Italy, 21–28.
- CASTELLI, D. AND PAGANO, P. 2002a. A flexible repository service: the OpenDLib solution. In *Proceedings of the 6th International ICCO/IFIP Conference on Electronic Publishing (ELPUB'02)*, J. A. Carvalho, A. Hübler, and A. A. Baptista, Eds. Verlag für Wissenschaft und Forschung, Berlin, Germany, 194–202.
- CASTELLI, D. AND PAGANO, P. 2002b. OpenDLib: A digital library service system. In *Proceedings of the 6th European Conference on Research and Advanced Technology for Digital Libraries (ECDL'02)*, M. Agosti and C. Thanos, Eds. Lecture Notes in Computer Science 2458, Springer, Heidelberg, Germany, 292–308.
- CLEVERDON, C. W. 1997. The Cranfield Tests on index languages devices. In *Readings in Information Retrieval*, K. Spack Jones and P. Willett, Eds. Morgan Kaufmann Publisher, Inc., San Francisco, CA, 47–60.
- CONSTANTOPOULOS, P., DOERR, M., THEODORIDOU, M., AND TZOBANAKIS, M. 2004. On information organization in annotation systems. In *Proceedings of the International Workshop on Intuitive Human Interfaces for Organizing and Accessing Intellectual Assets*, G. Grieser and Y. Tanaka, Eds. Lecture Notes in Computer Science, vol. 3359, Springer, Heidelberg, Germany, 189–200.

- DAVIS, J. R. AND HUTTENLOCHER, D. P. 1995. Shared annotation for cooperative learning. In *Proceedings of the 1st International Conference on Computer Support for Collaborative Learning (CSCL'95)*, J. L. Schnase and E. L. Cunnius, Eds. Lawrence Erlbaum Associates Inc., Mahwah, NJ, 84–88.
- DELOS. 2004. Newsletter—Issue 2. http://www.delos.info/index.php?option=com_content&task=view&id=392&Itemid=214 [last visited 2007, March 23].
- DIESTEL, R. 2000. *Graph Theory*. Springer-Verlag, New York.
- FERRO, N. 2004. Formal model and conceptual architecture of the annotation service for dynamic ubiquitous knowledge environments. Ph.D. thesis, Department of Information Engineering, University of Padua, Italy.
- FERRO, N. 2005. Design choices for a flexible annotation service. In *Post-Proceedings of the First Italian Research Conference on Digital Library Management Systems (IRCDL'05)*, M. Agosti and C. Thanos, Eds. ISTI-CNR at Gruppo ALI, Pisa, Italy, 101–110.
- FIELDING, R., GETTYS, Y., MOGUL, J., FRYSTYK, H., MASINTER, L., LEACH, P., AND BERNERS-LEE, T. 1999. Hypertext Transfer Protocol—HTTP/1.1. RFC 2616.
- FOGLI, D., FRESTA, G., AND MUSSIO, P. 2004. On electronic annotation and its implementation. In *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI '04)*, M. F. Costabile, Ed. ACM Press, New York, 98–102.
- FREED, N. AND BORENSTEIN, N. 1996a. *Multipurpose Internet Mail Extensions (MIME) Part Five: Conformance Criteria and Examples*. RFC 2049.
- FREED, N. AND BORENSTEIN, N. 1996b. *Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies*. RFC 2045.
- FREED, N. AND BORENSTEIN, N. 1996c. *Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types*. RFC 2046.
- FREED, N., KLENSIN, J., AND POSTEL, J. 1996. *Multipurpose Internet Mail Extensions (MIME) Part Four: Registration Procedures*. RFC 2048.
- FROMMHOLZ, I., BROCKS, H., THIEL, U., NEUHOLD, E., IANNONE, L., SEMERARO, G., BERARDI, M., AND CECI, M. 2003. Document-centered collaboration for scholars in the humanities—the COLLATE system. In *Proceedings of the 7th European Conference on Research and Advanced Technology for Digital Libraries (ECDL'03)*, T. Koch and I. T. Sølvsberg, Eds. Lecture Notes in Computer Science, vol. 2769, Springer, Heidelberg, Germany, 434–445.
- FROMMHOLZ, I., THIEL, U., AND KAMPS, T. 2004. Annotation-based document retrieval with four-valued probabilistic datalog. In *Proceedings of the 3rd XML and Information Retrieval Workshop and the 1st Workshop on the Integration of Information Retrieval and Databases (WIRD'04)*, R. Baeza-Yates, Y. Maarek, T. Roelleke, and A. P. de Vries, Eds. <http://homepages.cwi.nl/~arjen/wird04/wird04-proceedings.pdf> [last visited 2007, March 23], 31–38.
- GONÇALVES, M. A. AND FOX, E. A. 2002. 5SL—A Language for declarative specification and generation of digital libraries. In *Proceedings of the 2nd ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL '02)*, W. Hersh and G. Marchionini, Eds. ACM Press, New York, 263–272.
- GONÇALVES, M. A., FOX, E. A., WATSON, L. T., AND KIPP, N. A. 2004a. Streams, structures, spaces, scenarios, societies (5S): A formal model for digital libraries. *ACM Trans. Inform. Syst.* 22, 2 (Apr.), 270–312.
- GONÇALVES, M. A., WATSON, L. T., AND FOX, E. A. 2004b. Towards a digital library theory: A formal digital library ontology. In *ACM SIGIR Mathematical/Formal Methods in Information Retrieval Workshop (MF/IR 2004)*, S. Dominich and C. J. van Rijsbergen, Eds. http://www.dcs.vein.hu/CIR/mfir_2004.html [last visited 2007, March 23].
- GUEYE, B., RIGAUX, P., AND SPYRATOS, N. 2004. Taxonomy-based annotation of XML documents: Application to elearning resources. In *Proceedings of the 3rd Hellenic Conference on AI – Methods and Applications of Artificial Intelligence (SETN'04)*, G. A. Vouros and T. Panayiotopoulos, Eds. Lecture Notes in Computer Science 3025, Springer, Heidelberg, Germany, 33–42.
- HALASZ, F. G. 1988. Reflections on NoteCards: Seven issues for the next generation of hypermedia systems. *Comm. ACM* 31, 7, 836–852.
- HALASZ, F. G., MORAN, T. P., AND TRIGG, R. H. 1987. Notecards in a nutshell. In *Proceedings of the Conference on Human Factors in Computing Systems and Graphics Interface (CHI '87)*, J. M. Carroll and P. P. Tanner, Eds. ACM Press, New York, 45–52.

- HALMOS, P. R. 1974. *Naive Set Theory*. Springer-Verlag, New York.
- HANDSCHUH, S. AND STAAB, S., Eds. 2003. *Annotation for the Semantic Web*. IOS Press, Amsterdam, The Netherlands.
- HULL, D. 1993. Using statistical testing in the evaluation of retrieval experiments. In *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'93)*, R. Korfhage, E. Rasmussen, and P. Willett, Eds. ACM Press, New York, 329–338.
- IOANNIDIS, Y., MAIER, D., ABITEBOUL, S., BUNEMAN, P., DAVIDSON, S., FOX, E. A., HALEVY, A., KNOBLOCK, C., RABITTI, F., SCHEK, H.-J., AND WEIKUM, G. 2005. Digital library information-technology infrastructures. *Int. J. Dig. Lib.* 5, 4, 266–274.
- ISO. 2004. MPEG standards—coded representation of video and audio. <http://www.iso.ch/iso/en/prods-services/popstds/mpeg.html> [last visited 2007, March 23].
- JURAFSKY, D. AND MARTIN, J. H. 2000. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*. Prentice Hall, Upper Saddle River, NJ.
- KAHAN, J. AND KOIVUNEN, M.-R. 2001. Annotea: An open RDF infrastructure for shared Web annotations. In *Proceedings of the 10th International Conference on World Wide Web (WWW'01)*, V. Y. Shen, N. Saito, M. R. Lyu, and M. E. Zurko, Eds. ACM Press, New York, 623–632.
- KOIVUNEN, M.-R. AND SWICK, R. 2001. Metadata based annotation infrastructure offers flexibility and extensibility for collaborative applications and beyond. <http://www.w3.org/2001/Annotea/Papers/KCAP01/annotea.html> [last visited 2007, March 23].
- KOIVUNEN, M.-R., SWICK, R., AND PRUD'HOMMEAUX, E. 2003. *Annotea Shared Bookmarks*. <http://www.w3.org/2001/Annotea/Papers/KCAP03/annotaabm.html> [last visited 2007, March 23].
- KUNZE, J. 1995. *Functional Recommendations for Internet Resource Locators*. RFC 1736.
- LESK, M. 2005. *Understanding Digital Libraries*, 2nd ed. Morgan Kaufmann Publishers, San Francisco (CA), USA.
- MANN, W. C. AND THOMPSON, S. A. 1987. Rhetorical structure theory: A theory of text organization. Tech. Rep. ISI/RS-87-190, Information Sciences Institute, University of Southern California, Los Angeles, CA (June).
- MANNING, C. D. AND SCHÜTZE, H. 2001. *Foundations of Statistical Natural Language Processing*, 4th ed. The MIT Press, Cambridge, MA.
- MARSHALL, C. C. 1997. Annotation: from paper books to the digital library. In *Proceedings of the 2nd ACM International Conference on Digital Libraries (DL'97)*, R. B. Allen and E. Rasmussen, Eds. ACM Press, New York, 233–240.
- MARSHALL, C. C. 1998. Toward an ecology of hypertext annotation. In *Proceedings of the 9th ACM Conference on Hypertext and Hypermedia (HT'98): Links, Objects, Time and Space-Structure in Hypermedia Systems*, R. Akscyn, Ed. ACM Press, New York, 40–49.
- MARSHALL, C. C. AND BRUSH, A. J. B. 2002. From personal to shared annotations. In *Proceedings of the Conference on Human Factors and Computing Systems (CHI'02)—Extended Abstracts on Human Factors in Computer Systems*, L. Terveen and D. Wixon, Eds. ACM Press, New York, 812–813.
- MARSHALL, C. C. AND BRUSH, A. J. B. 2004. Exploring the relationship between personal and public annotations. In *Proceedings of the 4th ACM/IEEE-CS Joint Conference on Digital Libraries (JC'DL'04)*, H. Chen, H. Wactlar, C.-C. Chen, E.-P. Lim, and M. Christel, Eds. ACM Press, New York, 349–357.
- MARSHALL, C. C., GOLOVCHINSKY, G., AND PRICE, M. N. 2001a. Digital libraries and mobility. *Comm. ACM* 44, 55–56.
- MARSHALL, C. C., PRICE, M. N., GOLOVCHINSKY, G., AND SCHLIT, B. 1999. Introducing a digital library reading appliance into a reading group. In *Proceedings of the 4th ACM International Conference on Digital Libraries (DL'99)*, N. Rowe and E. A. Fox, Eds. ACM Press, New York, 77–84.
- MARSHALL, C. C., PRICE, M. N., GOLOVCHINSKY, G., AND SCHLIT, B. 2001b. Designing e-books for legal research. In *Proceedings of the 1st ACM/IEEE-CS Joint Conference on Digital Libraries (JC'DL'01)*, E. A. Fox and C. L. Borgman, Eds. ACM Press, New York, 41–48.
- MARSHALL, C. C. AND RUOTOLO, C. 2002. Reading-in-the-small: A study of reading on small form factor devices. In *Proceedings of the 2nd ACM/IEEE-CS Joint Conference on Digital Libraries (JC'DL'02)*, W. Hersh and G. Marchionini, Eds. ACM Press, New York, 56–64.

- MEALLING, M. AND DENENBERG, R. 2002. Report from the joint W3C/IETF URI Planning Interest Group: Uniform Resource Identifiers (URIs), URLs, and Uniform Resource Names (URNs): Clarifications and recommendations. RFC 3305.
- MOORE, K. 1996. Multipurpose Internet Mail Extensions (MIME) Part Three: Message header extensions for Non-ASCII Text. RFC 2047.
- NAGAO, K. 2003. *Digital Content Annotation and Transcoding*. Artech House, Norwood, MA.
- NAVARRO, G. AND BAEZA-YATES, R. 1997. Proximal nodes: A model to query document databases by content and structure. *ACM Trans. Inform. Syst.* 15, 4 (Oct.), 400–435.
- NEUHOLD, E., NIEDERÉE, C., STEWART, A., FROMMHOLZ, I., AND MEHTA, B. 2004. The role of context for information mediation in digital libraries. In *Proceedings of the 7th International Conference on Asian Digital Libraries. Digital Libraries: International Collaboration and Cross-Fertilization (ICADL'04)*, Z. Chen, H. Chen, Q. Miao, Y. Fu, E. A. Fox, and E.-P. Lim, Eds. Lecture Notes in Computer Science 3334, Springer, Heidelberg, Germany, 133–143.
- NISO. 2005. *ANSI/NISO Z39.88 - 2004—The OpenURL Framework for Context-Sensitive Services*. National Information Standards Organization (NISO). http://www.niso.org/standards/standard_detail.cfm?std_id=783 [last visited 2007, March 23].
- OAI. 2004. The Open Archives Initiative Protocol for Metadata Harvesting—Version 2.0. <http://www.openarchives.org/OAI/openarchivesprotocol.html> [last visited 2007, March 23].
- PASKIN, N., ED. 2006. *The DOI Handbook—Edition 4.4.1*. International DOI Foundation (IDF). <http://dx.doi.org/10.1000/186> [last visited 2007, March 23].
- PHELPS, T. A. AND WILENSKY, R. 1996. Towards active, extensible, networked documents: multivalent architecture and applications. In *Proceedings of the 1st ACM International Conference on Digital Libraries (DL'96)*, E. A. Fox and G. Marchionini, Eds. ACM Press, New York, 100–108.
- PHELPS, T. A. AND WILENSKY, R. 1997. Multivalent annotations. In *Proceedings of the 1st European Conference on Research and Advanced Technology for Digital Libraries (ECDL'97)*, C. Peters and C. Thanos, Eds. Lecture Notes in Computer Science 1324, Springer, Heidelberg, Germany, 287–303.
- PHELPS, T. A. AND WILENSKY, R. 2000a. Multivalent Documents. *Comm. ACM* 43, 6 (June), 83–90.
- PHELPS, T. A. AND WILENSKY, R. 2000b. Robust intra-document locations. *Computer Networks: The International Journal of Computer and Telecommunications Networking. Elsevier North-Holland, Inc., New York* 33, 1-6, 105–118.
- PHELPS, T. A. AND WILENSKY, R. 2001. The multivalent browser: A platform for new ideas. In *Proceedings of the 2001 ACM Symposium on Document Engineering*, E. V. Munson, Ed. ACM Press, New York, 58–67.
- RIGAUX, P. AND SPYRATOS, N. 2004. Metadata inference for document retrieval in a distributed repository. In *Proceedings of the 9th Asian Computing Science Conference—Advances in Computer Science (ASIAN '04)—Higher Decision Making. Dedicated to Jean-Louis Lassez on the Occasion of His 5th Cycle Birthday*, M. J. Maher, Ed. Lecture Notes in Computer Science, vol. 3321, Springer, Heidelberg, Germany, 418–436.
- SALTON, G. AND MCGILL, M. J. 1983. *Introduction to Modern Information Retrieval*. McGraw-Hill, New York.
- SCHILIT, B. N., PRICE, M. N., AND GOLOVCHINSKY, G. 1998. Digital library information appliances. In *Proceedings of the 3rd ACM International Conference on Digital Libraries (DL'98)*, I. Witten, R. Akscyn, and F. M. Shipman, Eds. ACM Press, New York, 217–226.
- STEIN, L. D., EDDY, S., AND DOWELL, R. 2002. Distributed Sequence Annotation System (DAS)—Version 1.53. <http://www.biodas.org/documents/spec.html> [last visited 2007, March 23].
- TAN, W.-C. 2004. Containment of relational queries with annotation propagation. In *Proceedings of the 9th International Workshop on Database Programming Languages (DBPL '03)*, G. Lausen and D. Suciu, Eds. Lecture Notes in Computer Science, vol. 2921, Springer, Heidelberg, Germany, 37–53.
- THIEL, U., BROCKS, H., FROMMHOLZ, I., DIRSCH-WEIGAND, A., KEIPER, J., STEIN, A., AND NEUHOLD, E. J. 2004. COLLATE—A collaboratory supporting research on historic European films. *Inter. J. Dig. Lib.* 4, 1 (Aug.), 8–12.
- VAN RIJSBERGEN, C. J. 1979. *Information Retrieval*, 2nd ed. Butterworths, London, England.

- W3C. 1998. Document Object Model (DOM) Level 1 Specification, Version 1.00—W3C Recommendation 1 October 1998. <http://www.w3.org/TR/REC-DOM-Level-1/> [last visited 2007, March 23].
- W3C. 1999. HTML 4.01 Specification—W3C Recommendation 24 December 1999. <http://www.w3.org/TR/html4/> [last visited 2007, March 23].
- W3C. 2005a. Annotea Project. <http://www.w3.org/2001/Annotea/> [last visited 2007, March 23].
- W3C. 2005b. EMMA: Extensible MultiModal Annotation markup language—W3C Working Draft 16 September 2005. <http://www.w3.org/TR/emma/> [last visited 2007, March 23].
- W3C. 2007. Semantic Web. <http://www.w3.org/2001/sw/> [last visited 2007, March 23].
- WITTEN, I. H. AND BAINBRIDGE, D. 2003. *How to Build a Digital Library*. Morgan Kaufmann Publishers, San Francisco, CA.
- XI, W., ZHANG, B., CHEN, Z., LU, Y., YAN, S., MA, W.-Y., AND FOX, E. A. 2004. Link fusion: A unified link analysis framework for multi-type interrelated data objects. In *Proceedings of the 13th International Conference on World Wide Web (WWW'04)*, S. Feldman, M. Uretsky, M. Najork, and C. Wills, Eds. ACM Press, New York, 319–327.

Received July 2005; revised March 2006, October 2006; accepted April 2007