Design Choices for a Flexible Annotation Service

N. Ferro

Department of Information Engineering
University of Padova

Abstract

This paper presents the main features of a flexible system capable of managing annotations in an automatic way in order to support users and their annotative practices, providing also advanced search functionalities based on annotations. Indeed, a flexible architecture allows the design of a system with a widespread usage, so that users can benefit from its functionalities without limitations due to the architecture of a particular system.

Keywords

H.3.7 Digital Libraries

1. Introduction

Nowadays, the notion of isolated information resources or applications is increasingly being replaced by a distributed and networked environment, where there is almost no distinction between local and remote information resources and applications. Indeed, a wide range of new technologies allow us to envision ubiquitous and pervasive access to information resources and applications. A wide range of wired and wireless technologies make it possible to offer almost ubiquitous connectivity; examples of such technologies are *Local Area Networks* (LANs), *Wireless LANs* (WLANs), *Asymmetric Digital Subscriber Line* (ADSL) and other broadband connections, *Third Generation Mobile System* (3G) networks as *Universal Mobile Telecommunication System* (UMTS) networks. Moreover, a variety of devices, that range from desktop computers to *Personal Digital Assistants* (PDAs), mobile phones, and other handheld devices [1], and a series of emerging architectural paradigms, such as *Web Services* (WS), *Peer-To-Peer* (P2P) and Grid architectures, are now available and allow us to design and develop services and systems that are more and more user-centered.

In particular, *Digital Libraries* (DLs), as information resources, and *Digital Library Management Systems* (DLMSs), that manage DLs, are currently in a state of evolution: today they are simply places where information resources can be stored and made available, whereas for tomorrow they will 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. This way, the user's intellectual work and the information resources provided by the DLMS can be merged together in order to constitute a single working context. Thus, the DL is no longer perceived as something external to the intellectual production process or as a mere consulting tool, but as an intrinsic and active part of the intellectual production process, as pointed out in [2, 3].

This turning point of DLMSs clearly emerges also from the outcomes of the third brainstorming meeting, organized by DELOS[3], the European Network of Excellence on Digital Libraries funded by the EU's 6th Framework Programme, which was held in Corvara, Italy on the 8–9 July, 2004[4]. The main conclusions were that: firstly, digital libraries have to become more user centred; secondly, digital libraries should not just be passive repositories but they should provide users with more active collaboration and communication tools; and thirdly, there is more of the need for generalized digital library management systems [4].

Annotations are effective means in order to enable this new paradigm of interaction between users and DLMSs, since they are a very well-established practice and widely used. 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 in order to improve collaborative work practices. Furthermore, annotations represent a bridge between reading and writing, that facilitates the user's first approach when they begin dealing with an information resource; thus, a DLMS offering annotation capabilities can be appealing to the user's needs. Finally, annotations allow users to naturally merge personal contents with the information resources provided by the DLMS, making it possible to embody the paradigm of interaction between users and DLs which has been envisaged above. We aim at designing a system capable of managing annotations in an automatic way in order to support users and their annotative practices, also providing advanced search functionalities based on annotations.

---

[3] `http://www.delos.info/`

[4] `http://www.delos.info/pastdelosevents.html`

The paper is organized as follows: Section 92 describes relevant characteristics of the annotations to be taken into account; Section 93 introduces our architectural approach for designing a flexible annotation service; Section 94 discusses how to exploit annotations for search purposes; Section 5 describes in detail the conceptual architecture of the system and how it supports advanced search functionalities based on annotations; Section 6 draws some conclusions.

2. Annotations

Over the years, a lot of research work concerning annotations has been done, where the main focus of this work has been on the employment of ad-hoc devices or handheld devices which enable reading appliances with annotation capabilities, or on the design and development of document models and systems which support annotations in specific management systems. All of this research work has led to different viewpoints about what an annotation is [3; 5]; these different viewpoints are taken into consideration in the following.

2.1. Annotations are metadata

They can be considered as additional data which concern an existing content, that is annotations are metadata, as they clarify in some way the properties and the semantics of the annotated content. For example, the Annotea[5] project developed by the *World Wide Web Consortium* (W3C) [6] sees annotations as metadata and interprets them as the first step in creating an infrastructure which will handle and associate metadata with content and will lead to the Semantic Web[6].

As a further example, *Multimedia Annotation of Digital Content Over the Web* (MADCOW) is based on a client-server architecture as Annotea is. Servers are repositories of annotations to which different client can connect, while the client is a plug-in for a standard Web browser [7, 8]. MADCOW employs *HyperText Transfer Protocol* (HTTP), in order to annotate Web resources and allows both private and public annotations. Moreover, it allows different pre-established types of annotations, such as explanation, comment, question, solution, summary, and so on.

Annotations are used also in the context of *DataBase Management Systems* (DBMSs) and, in particular, in the case of curated databases and scientific databases. SWISS-PROT[7] 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 domains structure, and so on. In this case, the annotations are embedded in the database and merged with the annotated content. BIODAS[8] provides a *Distributed Annotation System* (DAS), that is a Web–based server system 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, but they are separated from it. Moreover, [9, 10] investigate the usage of annotations with respect to the data provenance problem, which is the description of the origins of a piece of data and the process by which it arrived in a database, and [11] sees annotations as "information about data such as provenance, comments, or other types of metadata". Data provenance is a relevant issue in the field of curated and scientific databases, such as genome databases, because experts provide corrections and annotations to the original data, as time moves on.

2.2. Annotations are contents

Differently from the previous case, they are additional contents which concern an existing content [12]; indeed, they increase existing content by providing an additional layer of content that elucidates and explains the existing one. This viewpoint about annotations entails an intrinsic dualism between annotation as content enrichment and annotation as stand-alone document [13]:

- *annotation as content enrichment*: in this view annotations are considered as mere additional content regarding an existing document and as a result they are not autonomous entities but in fact they rely on previously existing information resources as to justify their existence;
- *annotation as stand-alone document*: in this view annotations are considered as real documents and are autonomous entities that maintain some sort of connection with an existing document.

This twofold nature of the annotation is clear if we think about the process of studying a document: firstly, we can start annotating some interesting passages that require an in-depth investigation, which is an annotation as content enrichment; then we can reconsider and collect our annotations and we can use them as a starting point for a new document, covering the points we would like to explain better, all of which is an 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.

[5] `http://www.w3.org/2001/Annotea/`
[6] `http://www.w3.org/2001/sw/`
[7] `http://www.expasy.org/sprot/`
[8] `http://biodas.org/`

2.3. Annotations constitute a hypertext
They allow the creation of new relationships among existing contents, by means of links that connect annotations together with existing content. In this sense we can consider that existing content and annotations constitute a hypertext [14], according to the definition of hypertext provided in [15]. This hypertext can be exploited not only for providing alternative navigation and browsing capabilities, but also for offering advanced search functionalities. Furthermore, [16] considers annotations as a natural way of creating and growing hypertexts that connect information resources in a DLMS by actively engaging users. Finally, the hypertext that exists between information resources and annotations enables different annotation configurations: the first are threads of annotations, i.e. an annotation made in response to another annotation, and the second are sets of annotations, i.e. a bundle of annotations on the same information resource [5, 13].

2.4. Annotations are dialog acts
They are part of a discourse with an existing content. For example, [17] considers annotations as the document context, intended as the context of the collaborative discourse in which the document is placed. Also [18] agree, to some extent, with this viewpoint about annotations. Indeed, they interpret annotations as a means that allow a "two way exchange of ideas between the authors of the documents and the documents users".

3. Architectural Approach
Annotations have a wide range of usages in different *Information Management Systems* (IMSs), ranging from DBMSs to DLMSs and corresponding to the different viewpoints about annotations, introduced in Section 2. Annotations are a key technology for actively involving users with an IMS and this technology should be available for each IMS employed by the user. Indeed, the user should benefit from a uniform way of interaction with annotation functionalities, without the need of changing their annotative practices only because a user works with different IMSs. Furthermore, annotations create an hypertext that allows users to merge their personal content with the information resources provided by diverse IMSs, according to the scenario envisaged in Section 0: this hypertext can span and cross the boundaries of a single IMS, if users need to interact with diverse IMSs. The possibility of having a hypertext that spans the boundaries of different IMSs is quite innovative because up to now such hypertext is usually confined within the boundaries of a single IMS. Moreover, IMSs do not usually offer hypertext management functionalities; for example, DLMSs do not normally have a hypertext connecting information resources with each other. Thus, annotations can be a way of associating a hypertext to a DL in order to enable an active and dynamic usage of information resources [5]. Finally, there are many new emerging architectural paradigms, such as P2P or WS architectures, that have to be taken into account.

Thus, our architectural approach is based on flexibility, because we need to adopt an architecture which is flexible enough to support both various architectural paradigms and a wide range of different IMSs. Indeed, a flexible architecture allows the design of a system with a widespread usage, so that users can benefit from its functionalities without limitations due to the architecture of a particular IMS. Since our target system is flexible, we named it *Flexible Annotation Service Tool* (FAST). In order to fulfil the requirements introduced above, our architectural approach is twofold:

1. to make FAST a stand-alone system, i.e. FAST is not part of any particular IMS;
2. to separate the core functionalities of the annotation service, from the functionalities needed to integrate it into different IMSs.

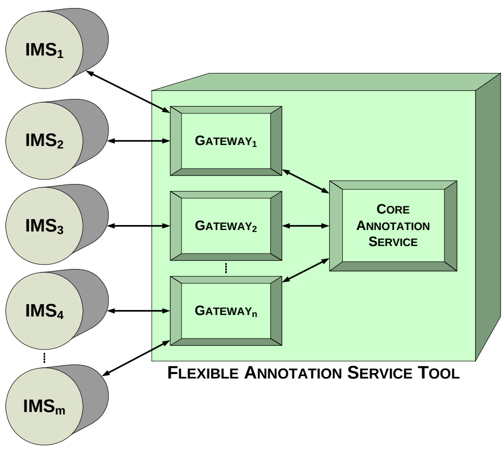

Figure 1: Overview of the architecture of FAST with respect to different IMSs.

Figure 1 shows the general architecture of the FAST system and its integration with different IMSs: the *Core Annotation Service* (CAS) is able to interact with different gateways, that are specialised for integrating the CAS into different IMSs. From the standpoint of an IMS the FAST system acts like any other distributed service of the IMS, even if it is actually made up of two distinct modules, the gateway and the CAS; on the other hand, the FAST system can be made available for another IMS by creating a new gateway. Note that the additional layer introduced by the gateway allows the integration of the CAS also with legacy systems, that may benefit from the availability of annotation functionalities.

The choice of making FAST a stand-alone system is coherent with the approach adopted by different systems: for example, Annotea by the W3C, MADCOW, and BIODAS rely on stand-alone servers, that store and manage annotations separated from the annotated objects. On the other hand, the choice of separating the core functionalities of the annotation service, from the functionalities needed to integrate it into the different IMSs is quite new. In fact, you will not be able to find an architecture like this in the literature about annotation systems, to the best of our knowledge.

As a consequence of this architectural choice, it is worth pointing out that the FAST system knows everything about annotations, however it cannot do any assumption regarding the information resources provided by the IMS, being that it needs to cooperate with different IMSs. This situation is very different from what is commonly found today. For example, both Annotea and MADCOW are stand-alone systems but they are targeted to work with Web pages. Indeed, they assume that the annotated object has a structured compliant with *HyperText Markup Language* (HTML), as an example, and that they can use HTTP to transport annotations. On the contrary, FAST cannot assume that it is dealing with either HTML documents or the HTTP protocol, but it has to avoid any constraints concerning both the annotated information resource and the available protocols. The only assumption about information resources that FAST can make is that each information resource is uniquely identified by a *handle*, which is a name assigned to an information resource in order to identify and facilitate the referencing to it, such as a *Uniform Resource Identifier* (URI) or a *Digital Object Identifier* (DOI).

4. Search Strategy Overview

Despite all of the research in modelling annotations and providing annotation–enabled systems, there is much less study regarding the usage of annotations for retrieving documents. [19] compares queries based on annotations with relevance feedback, and considers annotation–based queries as an automatic technique for query construction, since queries are automatically generated from annotated text, e.g. from highlighted text. [17] considers annotations – specifically annotations threads – as an extension of the document they belong to, creating a discourse context, in which not only the annotation itself but also its position in the discourse and its type, are exploited for searching and retrieving documents; this approach is revised and extended upon in [20] to probabilistic datalog. We need to develop a search strategy which is able to effectively take into account the multiple sources of evidence which come from both documents and annotations. In fact, the combining of these multiple sources of evidence can be exploited in order to improve the performances of an information management system. Our aim is to retrieve more documents that are relevant and to have them ranked in a way which is better than a system that does not makes use of annotations.

We will now introduce our search strategy by means of illustrating an example. Figure 2 shows a possible hypertext which could exist among documents and annotations, and which we have called document–annotation hypertext. Suppose that we have the following query: q = "`good survey grid computing`"

Firstly, we can start by searching the set of documents for this query. Let us suppose that we obtain the first result set $R_{d,q} = \{d_4, d_3\}$ ($R_{d,q}$ stands for: Result Documents by Query) where, intuitively, d_4 is ranked higher than d_3 because three query terms out of four are contained in d_4 while d_3 contains only two terms out of four. However, none of these two documents explains anything about how good the survey is and d_3 does not specify whether the document is a survey or not. Moreover, d_2 is not retrieved because it is concerned with computer networks in general and not with grid computing in particular.

Secondly, we can also search the set of annotations for this query. Suppose that we obtain the second result set $R_{a,q} = \{a_6, a_{12}, a_7\}$ ($R_{a,q}$ stands for: Result Annotations by Query) where, intuitively, a_6 has the highest rank because it contains all of the query terms; a_{12} is ranked lower than a_6 because it contains only two query terms; finally, a_7 has the lowest rank because it contains only one query term. It is worth noting that neither a_7 nor a_{12} explains what the topic of the survey is about, even if they provide additional information about the document they annotate; in a certain sense, it is the symmetric problem with respect to d_3 and d_4, that do not specify that much about the "survey side" of the query. At this point, we have two distinct sources of evidence on hand – the one which comes from the document set and the one which comes from the annotation set – and therefore we should exploit both of them in order to better satisfy the user's information need. Thus, we can exploit them with a twofold aim: firstly, to add new relevant documents to the result set and, secondly, to re-rank the documents in the result set.

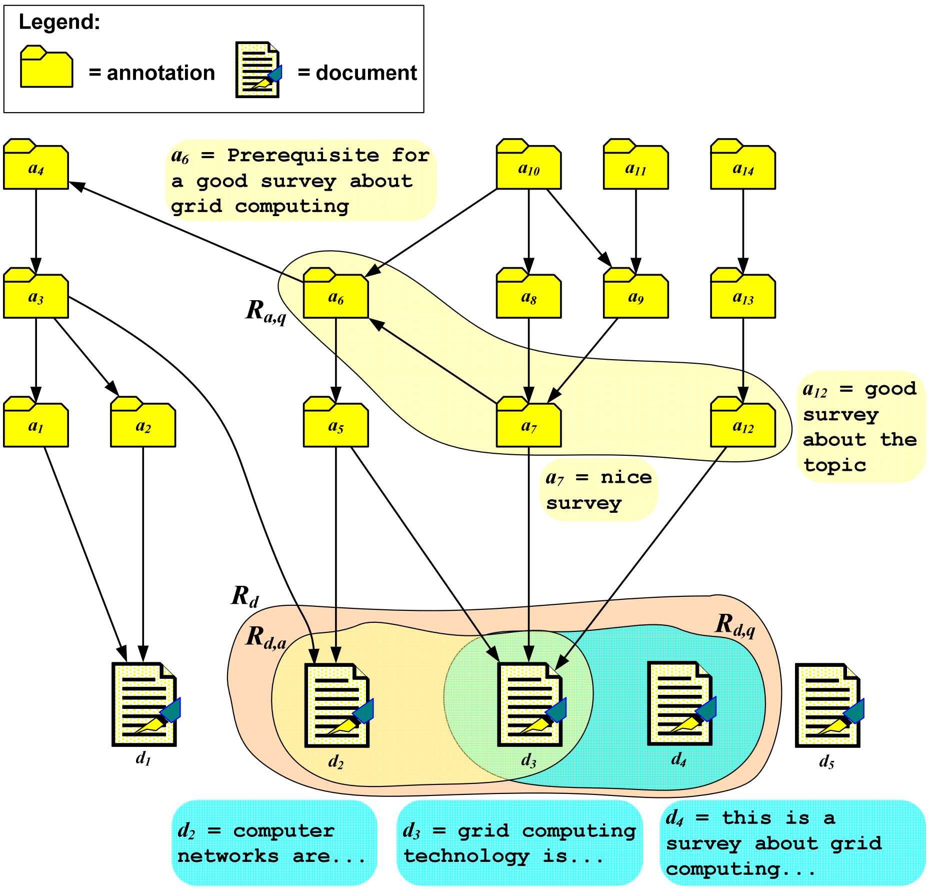

Figure 2: Example of the document–annotation hypertext used for search purposes.

With this in mind, we can note that:

- the annotations thread $a_6 \rightarrow a_5 \rightarrow d_2$ allows us to connect annotation a_6 to document d_2, suggesting that also document d_2 should be included in the result set. However, d_2 should not be ranked very high because, intuitively, it does not contain any query term and we deduce that it could be related to a survey about grid computing by means of an annotation that is two steps away from d_2;
- the annotations set a_7 and a_{12} regarding document d_3 allows us to understand that d_3 is a survey about grid computing, which is probably a good one. Therefore, we could consider ranking it higher.

Thus, we can identify a third result set $R_{d,a} = \{d_3, d_2\}$ ($R_{d,a}$ stands for: Result Documents by Annotation) where d_3 is ranked higher than d_2 for the reasons explained above. Note that we identified $R_{d,a}$ by means of $R_{a,q}$, that is we found the documents contained in $R_{d,a}$ using the annotations contained in $R_{a,q}$ and the document–annotation hypertext permitted us to pass from annotations ($R_{a,q}$) to documents ($R_{d,q}$).

We can conclude this line of reasoning with the final result set $R_d = \{d_3, d_4, d_2\}$ (R_d stands for: Result Documents). Intuitively, d_3 has the highest rank because it is strongly supported by its own evidence and the evidence provided by the annotations a_7 and a_{12}; in fact, $d_3 \in R_{d,q} \cap R_{d,a}$, as depicted in Figure . d_4 keeps its former rank, which is now lower than the rank given to d_3, due to the fact that it is not supported by any further evidence except its own; indeed, $d_4 \in R_{d,q} \setminus R_{d,a}$, as depicted in Figure . Finally, we add d_2 which has the lowest rank, due to the fact that it is supported only by the annotation a_6 which, as mentioned above, is not so close to d_2; indeed, $d_2 \in R_{d,a} \setminus R_{d,q}$, as depicted in Figure .

In conclusion, annotations provide us with an additional context which can be exploited with the ultimate goal of retrieving more documents that are relevant and better ranked. Furthermore, the document–annotation hypertext is the basic infrastructure which enables us to combine the sources of evidence which derive from documents and annotations. Thus, we face this research problem in the context of data fusion [21], because we need to combine the source of evidence which comes from annotations with the one which comes from documents. Moreover, also *Hypertext Information Retrieval* (HIR) techniques [22] are suitable in order to support the search strategy described above, because we need to deal with an hypertext in order to combine the different sources of evidence.

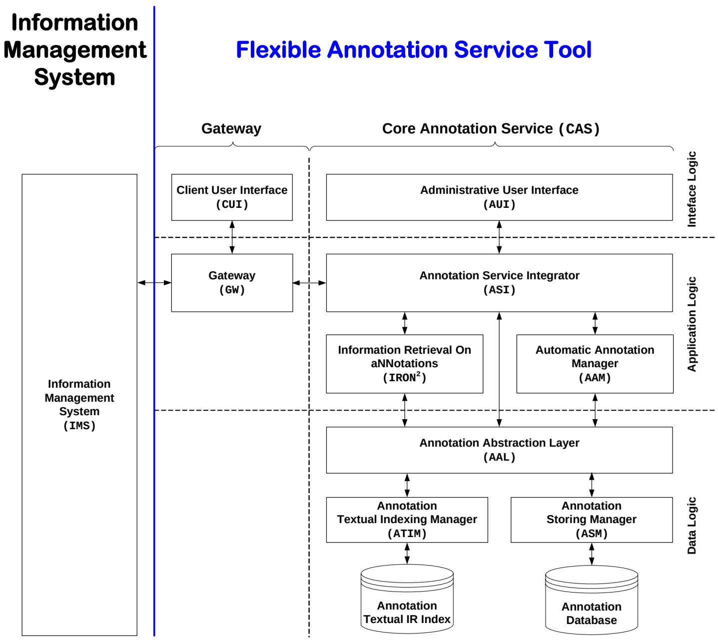

Figure 3: Detailed architecture of the FAST system.

5. FAST Conceptual Architecture

Figure 3 demonstrates the complete conceptual architecture of FAST, where FAST is depicted on the right, and the generic IMS is represented on the left. On the whole, the architecture is organized along two dimensions:

- *horizontal decomposition* (from left to right): consists of the IMS, the gateway and the CAS. It separates the core functionalities of FAST from the problem of integrating FAST into a specific IMS.
  The horizontal decomposition allows us to accomplish the first two requirements of our architecture, since FAST is a stand-alone system that can be integrated with different IMSs by changing the gateway;
- *vertical decomposition* (from bottom to top): consists of three layers – the data, application and interface logic layers – and it is concerned with the organization structure of the CAS.
  This decomposition allows us to achieve a better modularity within FAST and to properly describe the behaviour of FAST by means of isolating specific functionalities at the proper layer. Moreover, this decomposition makes it possibile to clearly define the functioning of FAST by means of communication paths that connect the different components of FAST itself. In this way, the behaviour of the FAST system is designed in a modular and extensible way.

The conceptual architecture of FAST is designed at a high level of abstraction in terms of abstract *Application Program Interfaces* (APIs) using an *Object Oriented* (OO) approach. In this way, we can model the behaviour and the functioning of FAST without worrying about the actual implementation of each component. Different alternative implementations of each component could be provided, still keeping a coherent view of the whole architecture of the FAST system. We achieve this abstraction level by means of a set of interfaces, which define the behaviour of each component of FAST in abstract terms. Then, a set of abstract classes partially implement the interfaces in order to define the actual behaviour common to all of the implementations of each component. Finally, the actual implementation is left to the concrete classes, inherited from the abstract ones, that fit FAST into a given architecture, such as aWS or a P2P architecture.

In the following sections we describe each component of FAST, according to Figure 3, from bottom to top.

5.1. Data Logic Layer

5.1.1. Annotation Storing Manager

The *Annotation Storing Manager* (ASM) manages the actual storage of the annotations and provides a persistence layer for storing the objects which represent the annotation and which are used by the upper layers of the architecture. The ASM relies on a *Relational DBMS* (RDBMS) in order to store annotations. The database schema is given by the mapping to the relational data model of the *Entity–Relationship* (ER) schema for modelling annotations, which has been proposed in [5, 23]. Thus, the ASM provides a set of basic operations for storing, retrieving, updating, deleting and searching annotations in a SQL–like fashion. Furthermore, it takes care of mapping the objects which represent the annotations into their equivalent representation in the relational model, according to the *Data Access Object* (DAO)[9] and the *Transfer Object* (TO)[9] design patterns. The DAO implements the access mechanism required to work with the underlying data source, i.e. it offers access to the RDBMS using the *Java DataBase Connectivity* (JDBC) technology. The components that rely on the DAO are called clients and they use the interface exposed by the DAO, which completely hides the data source implementation details from its clients. Because the interface exposed by the DAO to clients does not change when the underlying data source implementation changes, this pattern allows the DAO to adapt to different storage schemes without affecting its clients. Essentially, the DAO acts as an adapter between the clients and the data source. The DAO makes use of TOs as data carriers in order to return data to the client. The DAO may also receive data from the client in a TO in order to update the data in the underlying data source.

In conclusion, all of the other components of FAST deal only with objects representing annotations, which are the TOs of our system, without worrying about the details related to the persistence of such objects.

5.1.2. Annotation Textual Indexing Manager

The *Annotation Textual Indexing Manager* (ATIM) provides a set of basic operations for indexing and searching annotations for *Information Retrieval* (IR) purposes. The ATIM is a full-text *Information Retrieval System* (IRS) and deals with the textual content of an annotation. It is based on the experience acquired in developing *Information Retrieval ON* (IRON), the prototype IRS which has been used for participating in the *Cross-Language Evaluation Forum* (CLEF)[10] evaluation campaigns since 2002 [24].

5.1.3. Annotation Abstraction Layer

The *Annotation Abstraction Layer* (AAL) abstracts the upper layers from the details of the actual storage and indexing of annotations, providing uniform access to the functionalities of the ASM and the ATIM. The AAL provides the typical *Create–Read–Update–Delete* (CRUD) data management operations, coordinating the work of the ASM and the ATIM together. For example, when we create a new annotation, we need to put it into both the ASM and the ATIM. Furthermore, the AAL provides search capabilities by properly forwarding the queries to the ASM or to the ATIM. Our modular architecture allows us to partner the ATIM, which is specialised for providing full text search capabilities, with other IRSs, which are specialised for indexing and searching other kinds of media. In any case, the addition of other specialised IRSs is transparent for the upper layers, due to the fact that the AAL provides the upper layers with an uniform access to those IRSs.

Note that both the ASM and the ATIM are focused on each single annotation in order to properly store and index it. On the other hand, both the ASM and the ATIM do not have a comprehensive view of the relationships that exist between documents and annotations. On the contrary, the AAL has a global knowledge of the annotations and their relationships by using the hypertext existing between documents and annotations. For example, if we delete an annotation that is part of a thread of annotations, what policy do we need to apply? Do we delete all the annotations that refer to the deleted one or do we try to reposition those annotations? The ASM and the ATIM alone would not be able to answer this question but, on the other hand, the AAL can drive the ASM and the ATIM to perform the correct operations by exploiting the hypertext between documents and annotations.

5.2. Application Logic Layer

5.2.1. Automatic Annotation Manager

The *Automatic Annotation Manager* (AAM) automatically creates annotations for a given document. Automatic annotations can be created by using topic detection techniques in order to associate each annotation with its related topic, which constitutes the context of the annotation. In this way, a document can be re-organized and segmented into topics, whose dimension can range in many different sizes, and annotations can present a brief description of those topics.

5.2.2. Information Retrieval On aNNotations

The *Unified Modeling Language* (UML) sequence diagram of Figure shows how searching for documents by exploiting annotations involves many components of FAST. Remember that we aim at combining the source of evidence which comes from annotations, managed by FAST, with the source of evidence which comes from documents, managed by the IMS. Thus, the search strategy requires the cooperation of both FAST and the IMS in order to acquire these two sources of evidence. Firstly, FAST receives a query from the end-user, which is dispatched from the user interface to *Information Retrieval On aNNotations* ($IRON^2$). Secondly, the query is used to select all the relevant annotations, that is $IRON^2$ asks the *Annotation Service Integrator* (ASI) to find all

---

[9] `http://java.sun.com/blueprints/corej2eepatterns/Patterns/`

[10] `http://clef.isti.cnr.it/`

the relevant annotations. Then, the hypertext between documents and annotations can be built and used to identify the documents that are related to the found annotations. Now we aim to combine the source of evidence which comes from the documents identified by the annotations with the one which comes from the documents managed by the IMS, as previously explained. Since the source of evidence concerning the documents is completely managed by the IMS, the FAST system has to query the IMS, which gives us back a list of relevant documents. Finally, once the FAST system has acquired this information from the IMS, it can combine this information with the source of evidence which comes from the documents identified by annotations in order to create a list of fused result documents that are presented to the users.

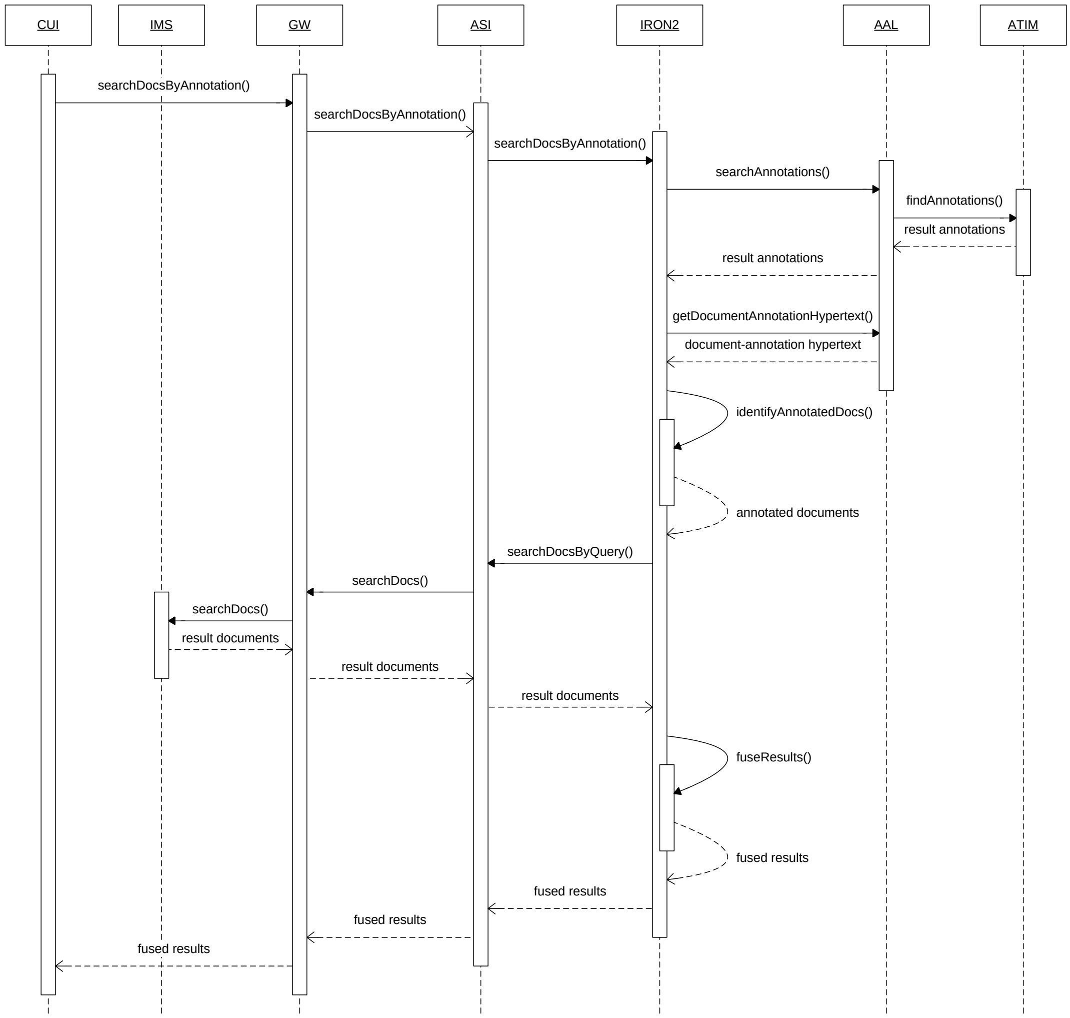

Figure 4: Sequence diagram for searching documents by exploiting annotations.

Our architectural choices influence the way in which our search strategy is carried out. Indeed, we aim at combining multiple sources of evidence which come from both documents and annotations. Since the source of evidence concerning the documents is completely managed by the IMS, FAST has to query the IMS in order to obtain it. Only after that FAST has acquired this information from the IMS, it can be combined with the source of evidence which comes from annotations in order to create a list of result documents that better satisfies the user's information needs. In conclusion, we deal with a distributed search problem.

5.2.3. Annotation Service Integrator

The ASI integrates the underlying components and provides uniform access to them. It represents the entry point to the CAS for both the gateway and the user interface, dispatching their requests to underlying layers and then collecting the responses from the underlying layers.

The UML sequence diagram of Figure 4 shows how the ASI plays a central role in coordinating the different components of FAST. In the example of Figure 4, the ASI forwards the user query to IRON^2; it dispatches the request for relevant documents of IRON^2 to the *Gateway* (GW) in order to submit this query to the IMS; then, it passes the results provided by the IMS back to IRON^2; finally, it gives the fused result list produced by IRON^2 back to the GW in order to return this list to the user interface.

5.2.4. Gateway
As already discussed in Section 3, the GW provides functionalities of mediator between the CAS and the IMS. By changing the gateway, we can share FAST with different IMSs. In this way, we can provide a wide range of different architectural choices: firstly, the CAS could be connected to a specific IMS which uses proprietary protocols and data structures and, in this case, the gateway can implement them; secondly, we could employ WS to carry out the gateway, so that FAST is accessible in a more standardized way; finally, the gateway could be used to adapt FAST to a P2P network of IMSs.

5.3. Interface Logic Layer

5.3.1. Administrative User Interface
The *Administrative User Interface* (AUI) is a Web-based UI for the administration of FAST. It provides the different functionalities needed to configure and run FAST, such as the choice of the gateway to be used, the creation and management of the users granted by the system, and so on.

5.3.2. Client User Interface
The *Client User Interface* (CUI) provides end–users with an interface for creating, modifying, deleting and searching annotations. The CUI is connected to, or even directly integrated into, the gateway, so that it represents a user interface tailored to the specific IMS for which the gateway is developed. In this way, the gateway forwards the requests from the CUI to the ASI, as it is shown in the example of Figure 4.

6. Conclusions
This paper discussed the conceptual architecture of the FAST system, which separates core functionalities from their integration in any particular IMS. In this way, FAST acts as a bridge between different IMSs and allows the hypertext to cross the boundaries of a single IMS, in order to exploit annotations as an active and effective collaboration tool for users.

Acknowledgements
The work reported in this paper has been conducted in the context of a joint program between the Italian National Research Council (CNR) and the Ministry of Education (MIUR), under the law 449/97-99. The work is also 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).

References
[46] M. Agosti and N. Ferro, "Chapter X: Managing the Interactions between Handheld Devices, Mobile Applications, and Users" in E.P. Lim and K. Siau, eds., "Advances in Mobile Commerce Technologies," Idea Group, Hershey USA, 2003, pp. 204–233.

[47] M. Agosti and N. Ferro, "An Information Service Architecture for Annotations" in M. Agosti, H.-J. Schek, and C. Türker, eds, "Digital Library Architectures: Peer-to-Peer, Grid, and Service-Orientation, Pre-proceedings of the 6th Thematic Workshop of the EU Network of Excellence DELOS", Edizioni Libreria Progetto, Padova Italy, 2004, pp. 115–126.

[48] M. Agosti and N. Ferro, "A System Architecture as a Support to a Flexible Annotation Service" in C. Türker, M. Agosti, and H.-J. Schek, eds, "Peer-to-Peer, Grid, and Service-Orientation in Digital Library Architectures: 6th Thematic Workshop of the EU Network of Excellence DELOS. Revised Selected Papers", Lecture Notes in Computer Science (LNCS) 3664, Springer, Heidelberg Germany, 2005, pp. 147–166.

[49] DELOS, Newsletter – Issue 2. `http://www.delos.info/newsletter/issue2/`, October 2004 [last visited 2005, June 24].

[50] M. Agosti, N. Ferro, I. Frommholz, and U. Thiel, "Annotations in Digital Libraries and Collaboratories – Facets, Models and Usage", in R. Heery and L. Lyon, eds, "Proc. 8th European Conference on Research and Advanced Technology for Digital Libraries (ECDL 2004)", Lecture Notes in Computer Science (LNCS) 3232, Springer, Heidelberg Germany, 2004, pp. 244–255.

[51] J. Kahan and M.-R. Koivunen, "Annotea: an open RDF infrastructure for shared Web annotations", in V. Y. Shen, N. Saito, M. R. Lyu, and M. E. Zurko, eds, "Proc. 10th International Conference on World Wide Web (WWW 2001)", ACM Press, New York USA, 2001, pp 623–632.

[52] P. Bottoni, R. Civica, S. Levialdi, L. Orso, E. Panizzi, and R. Trinchese, "MADCOW: a Multimedia Digital Annotation System", in M.F. Costabile, ed., "Proc. Working Conference on Advanced Visual Interfaces (AVI 2004)", ACM Press, New York USA, 2004, pp. 55–62.

[53] P. Bottoni, R. Civica, S. Levialdi, L. Orso, E. Panizzi, and R. Trinchese, "Storing and Retrieving Multimedia Web Notes", in S. Bhalla, ed., "Proc. 4th International Workshop on Databases in Networked

Information Systems (DNIS 2005)", Lecture Notes in Computer Science (LNCS) 3433, Springer, Heidelberg Germany, 2005, pp. 119–137.
[54] P. Buneman, S. Khanna, W.C. Tan, "Why and Where: A Characterization of Data Provenance", in J. Van den Bussche and V. Vianu, eds., "Proc. 8th International Conference on Database Theory (ICDT 2001)", Lecture Notes in Computer Science (LNCS) 1973, Springer, Heidelberg Germany, 2001, pp. 316–330.
[55] P. Buneman, S. Khanna, W.C. Tan, "On Propagation of Deletions and Annotations Through Views", in S. Abiteboul, P.G. Kolaitis, and L. Popa, eds., "Proc. 21st ACM SIGMOD–SIGACT–SIGART Symposium on Principles of Database Systems (PODS 2002)", ACM Press, New York USA, 2002, pp. 150–158.
[56] D. Bhagwat, L. Chiticariu, W.-C. Tan, and G. Vijayvargiya, "An Annotation Management System for Relational Databases", in M. A. Nascimento, M. T. Özsu, D. Kossmann, R. J. Miller, J. A. Blakeley, and K. B. Schiefer, eds, "Proc. 30th International Conference on Very Large Data Bases (VLDB 2004)", Morgan Kaufmann, 2004, pp. 900–911.
[57] K. Nagao, "Digital Content Annotation and Transcoding", Artech House, Norwood (MA)USA, 2003.
[58] M. Agosti and N. Ferro, "Annotations: Enriching a Digital Library", in T. Koch and I. T. Sølvberg, eds, "Proc. 7th European Conference on Research and Advanced Technology for Digital Libraries (ECDL 2003)", Lecture Notes in Computer Science (LNCS) 2769, Springer, Heidelberg Germany, 2003, pp. 88–100.
[59] M. Agosti and N. Ferro, "Annotations as Context for Searching Documents", in F. Crestani and I. Ruthven, eds, "Proc. 5th International Conference on Conceptions of Library and Information Science – Context: nature, impact and role", Lecture Notes in Computer Science (LNCS) 3507, Springer, Heidelberg Germany, 2005, pp. 155–170.
[60] M. Agosti, "An Overview of Hypertext", in M. Agosti and A. Smeaton, eds, "Information Retrieval and Hypertext", Kluwer Academic Publishers, Norwell (MA) USA, 1996, pp. 27–47.
[61] C. C. Marshall, "Toward an Ecology of Hypertext Annotation", in R. Akscyn, ed, "Proc. 9th ACM Conference on Hypertext and Hypermedia (HT 1998): links, objects, time and space-structure in hypermedia systems", ACM Press, New York USA, 1998, pp. 40–49.
[62] I. Frommholz, H. Brocks, U. Thiel, E. Neuhold, L. Iannone, G. Semeraro, M. Berardi, and M. Ceci, "Document-Centered Collaboration for Scholars in the Humanities – The COLLATE System", in T, Koch and I. T. Sølvberg, eds, "Proc. 7th European Conference on Research and Advanced Technology for Digital Libraries (ECDL 2003), Lecture Notes in Computer Science (LNCS) 2769, Springer, Heidelberg Germany, 2003, pp 434–445.
[63] D Fogli, G. Fresta, and P. Mussio, "On Electronic Annotation and Its Implementation" in M. Costabile, ed, "Proc. Working Conference on Advanced Visual Interfaces (AVI 2004)", ACM Press, New York USA, 2004, pp. 98–102.
[64] G. Golovchinsky, M.N. Price, and B.N. Schilit, "From Reading to Retrieval: Freeform Ink Annotations as Queries", in F. Gey, M. Hearst, and R. Tong, eds., "Proc. 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1999)", ACM Press, New York USA, 1999, pp. 19–25.
[65] I. Frommholz, U. Thiel, and T. Kamps, "Annotation-based Document Retrieval with Four-Valued Probabilistic Datalog", in R. Baeza-Yates, Y. Maarek, T. Roelleke, and A.P. de Vries, eds., Proc. 3rd XML and Information Retrieval Workshop and the 1st Workshop on the Integration of Information Retrieval and Databases (WIRD2004)", `http://homepages.cwi.nl/~arjen/wird04/wird04-proceedings.pdf`, 2004 [last visited 2005, April 5] , pp. 31–38.
[66] W.B. Croft, "Combining Approaches to Information Retrieval", in W.B. Croft, ed., "Advances in Information Retrieval: Recent Research from the Center for Intelligent Information Retrieval", Kluwer Academic Publishers, Norwell (MA) USA, 2000, pp. 1–36.
[67] M. Agosti and A. Smeaton, eds, "Information Retrieval and Hypertext", Kluwer Academic Publishers, Norwell (MA) USA, 1996.
[68] M. Agosti, N. Ferro, and N. Orio, "Annotating Illuminated Manuscripts: an Effective Tool for Research and Education", in M. Marlino, T. Sumner, and F. Shipman, eds, "Proc. 5th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL 2005)", ACM Press, New York USA, 2005, pp. 121–130.
[69] G.M. Di Nunzio, N. Ferro, and N. Orio, "Experiments on Statistical Approaches to Compensate for Limited Linguistic Resources", in C. Peters, P. Clough, J. Gonzalo, G. J. F. Jones, M. Kluck, and B. Magnini, eds, Multilingual Information Access for Text, Speech and Images: Fifth Workshop of the Cross--Language Evaluation Forum (CLEF 2004) Revised Selected Papers, Lecture Notes in Computer Science (LNCS) 3491, Springer, Heidelberg Germany, 2005, pp. 60–72.