Integration of the DiLAS Annotation Service into Digital Library Infrastructures

Maristella Agosti
University of Padua
Padua, Italy
agosti@dei.unipd.it

Nicola Ferro
University of Padua
Padua, Italy
ferro@dei.unipd.it

Ingo Frommholz
University of Duisburg-Essen
Duisburg, Germany
ingo.frommholz@uni-due.de

Emanuele Panizzi
University of Rome “La Sapienza”
Rome, Italy
panizzi@uniroma1.it

Wolfgang Putz
Fraunhofer IPSI
Darmstadt, Germany
putz@ipsi.fhg.de

Ulrich Thiel
Fraunhofer IPSI
Darmstadt, Germany
thiel@ipsi.fhg.de

ABSTRACT

In this paper we present DiLAS, a dedicated service for the management of annotations, supporting complex collaborative applications in distributed environments. The mission of DiLAS is to foster change in users' interaction with digital libaries (DL) and contribute to developing services for social infrastructures in DLs. We thus introduce a new independent annotation service for interactive knowledge creation and sharing, ready to be plugged into existing DL systems. As a show case, we discuss the integration of the DiLAS annotation service into DAFFODIL, an existing digital library management framework. As a future show case, the service is going to be integrated into the BRICKS infrastructure.

Categories and Subject Descriptors: H.3.5 [Information Storage and Retrieval]: Digital Libraries

General Terms: Design, Management

Keywords: Annotations, Annotation Service, DiLAS, DAFFODIL, Digital Libraries, DLMS, FAST, MADCOW

1. INTRODUCTION

In most contemporary digital library management systems (DLMS) the contents are conveyed to the user as a “collection of information items”, which can be searched or browsed. However, this paradigm is often not sufficient to cope with embedded usages, for which access to the contents is not seen as an isolated activity, but as part of a larger work process, where interaction with other users, editing and annotating documents need to be integrated. Up to now, annotations have been – in most cases – stored together with the documents they refer to in a central DL repository. However, the treatment of annotations as another type of contents often fails to meet the requirements of more complex applications, where annotations are employed as a means to establish communicative and collaborative functions for a distributed user community [2]. Therefore, a dedicated approach to modeling and management of annotations is needed, especially when complex collaborative applications have to be implemented in distributed environments. With the advent of decentralized DL architectures in Grid or Peer-to-Peer environments, but also in Service-oriented architectures, design choices need to be revised by technical solutions that allow us to manage annotations independently from a particular DLMS. To this end, the authors are participating in a research project, named Digital Library Annotation Service (DiLAS), aimed at designing and developing an architecture and a framework able to support and evaluate such a decentralized annotation service. The considerations presented here are based on the several experiences the authors gained by developing a number of annotation systems (like *BRICKS* [8], *COLLATE* [9], *DAFFODIL* [6], *FAST* [1], *IPSA* [3] and *MADCOW* [4]) targeted to different application domain users. Our experiences gained with designing and implementing these systems influenced the design decisions made for the DiLAS prototype.

In the next section we discuss objectives and benefits of creating an independent and flexible annotation service. In Section 3 the DiLAS annotation service is introduced. This service has been implemented and integrated into DAFFODIL, which is subsequently described in Section 4. In Section 5 we continue with a discussion of future work, in particular the integration of the DiLAS annotation service into BRICKS (Building Resources for Integrated Cultural Knowledge Services) [8], which is an EU funded Integrated Project (IST 507457) aiming to design, develop and maintain an open user and service-oriented infrastructure to share knowledge and resources in the Cultural Heritage domain; its target audience is very broad and heterogeneous and involves cultural heritage and educational institutions, research community, industry, and citizens.

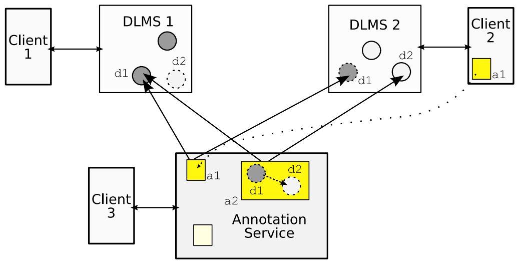


Figure 1: An independent annotation service

2. THE INDEPENDENT ANNOTATION SERVICE

2.1 Objectives

Our research work represents an effort in the design and development of software systems which are able to provide annotation capabilities on the content that they manage, and in the integration of such functionality in Digital Libraries (DL). The mission of DiLAS is to foster changes in users' interaction with DLs and contribute to developing services for social infrastructures in DLs. Users should be supported in performing their various collaborative tasks. It is therefore crucial that an annotation service aids users in creating and sharing new ideas and thoughts among their relevant community. DiLAS' mission is addressed through the provision of a new independent annotation service for interactive knowledge creation and sharing, ready to be plugged into existing DL systems. It is DiLAS' vision that the new annotation service inspires the users to become active and visible contributors with new knowledge. The annotation service enriches the DL contents, the users personalize the information in a new contextual learning opportunity, and they collaborate by sharing this new knowledge beyond the borders of a specific digital library. Consider the situation depicted in Figure 1, where a generic and independent annotation service is integrated into several digital library management systems. Users access and use the annotation service through the clients of a DLMS or directly (as it is the case for client 3). An organization might decide to make (parts of) their managed objects accessible for distributed use, as it the case in the example with documents `d1` (managed by DLMS 1) and `d2` (managed by DLMS 2), which are visible in other systems as well. `d1` might be annotated through a client of DLMS 2, and the resulting annotation `a1` can be visible through DLMS 1 and other clients to the annotation service as well. Another annotation `a2` might even relate `d1` and `d2`, creating a logical hypertext across the borders of a single DLMS. An organization collecting documents still remains responsible for its own data, while the corresponding annotations, which might be the property of other persons using a different DLMS and client, are organized by an independent service.

2.2 Benefits

As briefly outlined above, the DiLAS annotation service supports many generic tasks involving annotations, which makes it a candidate to be integrated into different DL systems (an example scenario is presented in Section 4). From a usability standpoint, a running DiLAS server, integrated into different DL systems, allows for using different clients for reading, searching and creating new annotations. As an example, one user might prefer a sophisticated user client, while another one is satisfied with a simple Web frontend running in his browser. Both users could access the same documents and annotations with their preferred client.

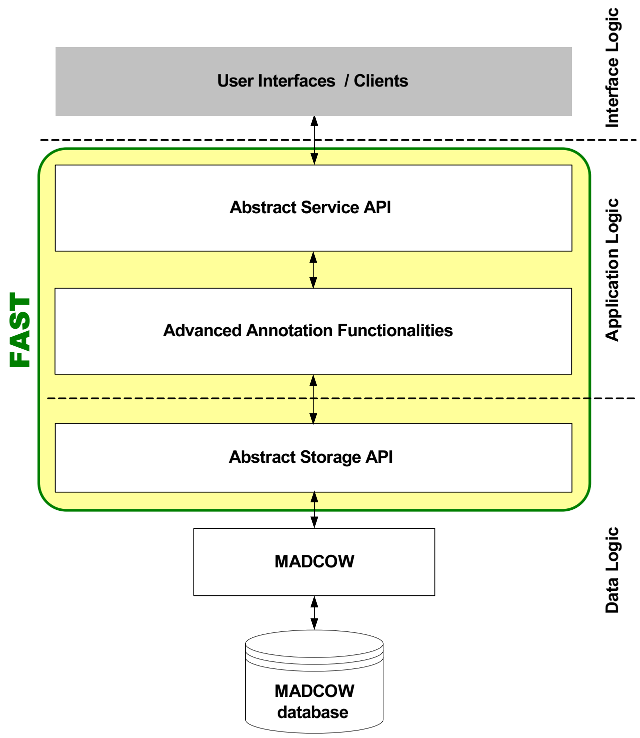


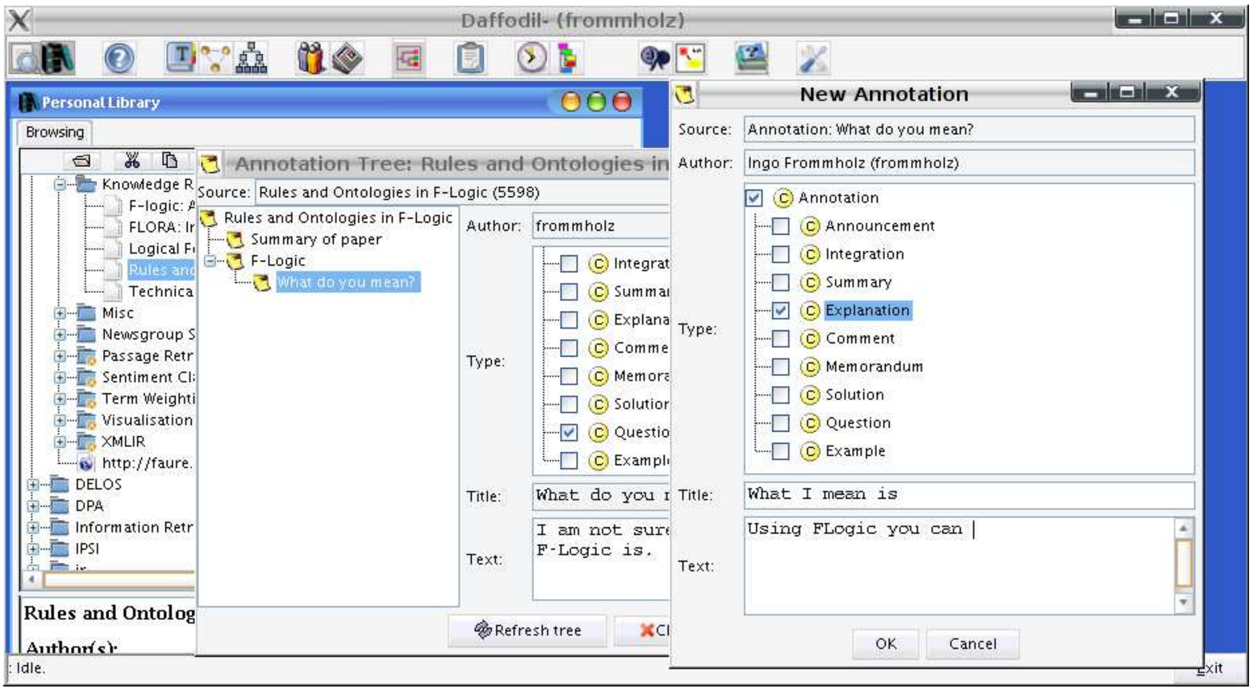
Figure 2: DiLAS Architecture

Another benefit of the generic DiLAS concept is that we are not bound to a specific document type; even non-digital or hybrid collections can use DiLAS, as long as these repositories address their content using uniform resource identifiers (URIs).

3. DILAS ANNOTATION SERVICE

3.1 DiLAS System Architecture

As already discussed, one of the goals of our project is to design and develop a generic annotation service, that is a service that can be easily used by different DLMS. To this end, the architecture of the DiLAS system, shown in Figure 2, consists of three layers – the data, application and interface logic layers. This decomposition allows us to achieve a better modularity within DiLAS and to properly describe the behaviour of DiLAS by means of isolating specific functionalities at the proper layer. The *data logic layer* 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. In order to make it as flexible as possible, an abstract API for the functionalities of the storage has been defined. This API, in turn, allows us for accessing different system to perform the actual storage of the annotations. In the first prototype of the DiLAS system we use the MADCOW system



Source: Rules and Ontologies in F-Logic (5598)

Author: frommholz

Type: □ Integrat □ Summar □ Explana □ Comme □ Memora □ Solution ☑ Questio □ Exampl

Title: What do you r

Text: I am not sure F-Logic is.

Source: Annotation: What do you mean?

Author: Ingo Frommholz (frommholz)

Type: ☑ Annotation
- □ Announcement
- □ Integration
- □ Summary
- ☑ Explanation
- □ Comment
- □ Memorandum
- □ Solution
- □ Question
- □ Example

Title: What I mean is

Text: Using FLogic you can |

Figure 3: DAFFODIL user interface with annotation functionality

as actual storage for the annotations.

The *application logic layer* provides advanced functionalities that make use of annotations. As in the case of the data logic layer, we defined a set of abstract API that make the access to the DiLAS service functionalities independent from the particular implementation provided.

The *interface logic layer* is devoted to manage the interaction with the end user. It depends on the system into which DiLAS is going to be used and relies on the DiLAS Abstract Service API in order to provide functionalities to the end user.

Both the application logic (Abstract Service API) and the upper part of the data logic (Abstract Storage API) correspond to the respective layers of the FAST system [1]. Indeed, FAST describes both these layers and the business objects exchanged among these layers by means of abstract interfaces. Those interfaces provide us with a general framework for describing the interaction and integration of the different layers without coupling it with a specific implementation. As a consequence, the integration of MADCOW requires to provide a concrete subclass of the FAST layers and business objects, in order to fit them to the needs of the newly integrated systems.

3.2 Integration of FAST and MADCOW

The proposed Abstract Storage API, see figure 2, has been implemented by creating a subclass of `FastDatastore`, called `FastMadcowDatastore`, which provides an implementation of the FAST `Datastore` interface, able to communicate with the MADCOW system. MADCOW is a Web-based system which communicates with its own protocol over HTTP. Thus, `FastMadcowDatastore` translates the functionalities described by the Abstract Storage API interface into the HTTP calls supported by MADCOW and exchanges with it XML encoded messages, according to the XML schemas defined for MADCOW. To this end, `FastMadcowDatastore` relies on the *Jakarta Commons HttpClient*[1] version 3.0, in order to hide the details of the communication with MADCOW over HTTP.

4. SHOW CASE: THE DAFFODIL/DILAS INFRASTRUCTURE

DAFFODIL [6] is targeted at the support of the *digital library life cycle* proposed by Paepcke [7]. While initially focussing on *strategic retrieval support*, improvements of DAFFODIL concentrate on *interpreting* the material at hand, *sharing* new insights and *creating* new knowledge. As annotations have been identified as playing a crucial part in all these aspects [2], we integrated the DiLAS Annotation Service into DAFFODIL, in order to provide the user with support for basic annotation operations like the creation of annotations, browsing of annotation threads and display of particular annotations. As an example for the various possible scenarios supported by the DAFFODIL/DiLAS integrated prototype, users can discuss the content of important documents in their state-of-the-art with collaborators in order to develop new ideas based on previous ones; these ideas can later result in new publications which in turn become part of a DL system's repository.

Figure 3 shows a screenshot of the DAFFODIL user interface. In DAFFODIL, users can categorise relevant documents in a personal library (PLib) using folders and subfolders. In the screenshot we can see the PLib on the left; a document was selected and its corresponding annotation thread (containing public annotations) is shown in the middle window. The right window shows the user interface for

[1] `http://jakarta.apache.org/commons/httpclient/`

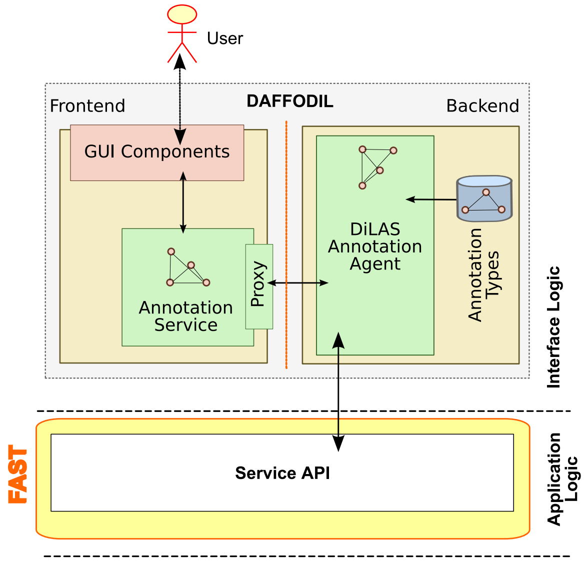


Figure 4: DAFFODIL/DiLAS Architecture

creating a new annotation (in this case an explanation of the previous question). Users can give a title for the annotation, a text, and can select a suitable annotation type.

Figure 4 shows the architecture of the DAFFODIL/DiLAS infrastructure. DAFFODIL consists of two main parts: the *backend*, running on a dedicated server, and the *frontend* executed on the user's machine. The backend consists of a set of agents establishing the functionality needed for the frontend services provided to the user. A DiLAS annotation agent was implemented and is running in the DAFFODIL back-end. This agent communicates with the FAST server. On the DAFFODIL frontend, an annotation service is running which communicates with the DAFFODIL backend through a proxy. To process annotation-related requests coming from the GUI components, the frontend annotation service propagates these requests to the DiLAS annotation agent in the backend through a proxy.

Annotations in DAFFODIL are modelled as an ontology. In this annotation ontology, each annotation type is an own class inheriting from the generic class Annotation. This taxonomy is displayed in the annotation windows in figure 3. Annotations themselves are instances of the classes of this ontology. For example, annotations of type "comment" are instances of the Comment class. For the integration of DiLAS into DAFFODIL, we developed a new annotation ontology containing a class for each annotation type supported by DiLAS.

5. FUTURE WORK

To illustrate the viability of our flexible annotation service approach, DiLAS will also be integrated into the BRICKS P2P network. The main goal of the BRICKS annotation service is to support different applications in facilitating collaboration and fostering the interpretation as well as the creation of new resources. To cover a broad range of applications, different types of annotations (text annotation, structured annotation, association) are supported, which allow to annotate any object or parts of it. Semantic labeling of annotations and a three layer access model (for private, collective or public access) supplement the model. The model has its roots in the COLLATE system, but extends it by a more generic approach to be usable in a service-oriented system. Integrating DiLAS into BRICKS should be understood as more than a proof of concept of the DiLAS architecture. The integration opens new possibilities to use the most appropriate annotation model for an application leaving open which persistent storage is used.

6. ACKNOWLEDGEMENTS

The 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).

7. REFERENCES

[1] M. Agosti and N. Ferro. A system architecture as a support to a flexible annotation service. In *Peer-to-Peer, Grid, and Service-Orientation in Digital Library Architectures: 6th Thematic Workshop of the EU Network of Excellence DELOS. Revised Selected Papers*, number 3664 in LNCS, pages 147–166, Heidelberg et al., 2005. Springer.

[2] M. Agosti, N. Ferro, I. Frommholz, and U. Thiel. Annotations in digital libraries and collaboratories – facets, models and usage. In Heery and Lyon [5], pages 244–255.

[3] M. Agosti, N. Ferro, and N. Orio. Annotating illuminated manuscripts: an effective tool for research and education. In *JCDL '05: Proceedings of the 5th ACM/IEEE-CS Joint Conference on Digital Libraries*, pages 121–130, New York, NY, USA, 2005. ACM Press.

[4] P. Bottoni, R. Civica, S. Levialdi, L. Orso, E. Panizzi, and R. Trinchese. Madcow: a multimedia digital annotation system. In *AVI '04: Proceedings of the working conference on Advanced visual interfaces*, pages 55–62, New York, NY, USA, 2004. ACM Press.

[5] R. Heery and L. Lyon, editors. *Research and Advanced Technology for Digital Libraries. Proc. European Conference on Digital Libraries (ECDL 2004)*, Lecture Notes in Computer Science, Heidelberg et al., 2004. Springer.

[6] C.-P. Klas, N. Fuhr, and A. Schaefer. Evaluating strategic support for information access in the DAFFODIL system. In Heery and Lyon [5].

[7] A. Paepcke. Digital libraries: Searching is not enough–what we learned on-site. *D-Lib Magazine*, 2(5), May 1996. `http://www.dlib.org/dlib/may96/stanford/05paepcke.html`.

[8] T. Risse, P. Knezevic, C. Meghini, R. Hecht, and F. Basile. The BRICKS infrastructure - an overview. In *Proc. of 75th Conference on Electronic Imaging, the Visual Arts & Beyond (EVA 2005)*, November/December 2005.

[9] U. Thiel, H. Brocks, I. Frommholz, A. Dirsch-Weigand, J. Keiper, A. Stein, and E. Neuhold. COLLATE - a collaboratory supporting research on historic european films. *International Journal on Digital Libraries (IJDL)*, 4(1):8–12, 2004.