DelosDLMS – the Integrated
DELOS Digital Library Management System

Maristella Agosti[1], Stefano Berretti[2], Gert Brettlecker[3], Alberto del Bimbo[2], Nicola Ferro[1], Norbert Fuhr[4], Daniel Keim[5], Claus-Peter Klas[4], Thomas Lidy[6], Moira Norrie[7], Paola Ranaldi[3,8], Andreas Rauber[6], Hans-Jörg Schek[8], Tobias Schreck[5], Heiko Schuldt[3], Beat Signer[7], Michael Springmann[3]

[1] Information Systems Management Research Group, University of Padova, Italy
[2] Visual Information Processing Laboratory, University of Florence, Italy
[3] Database and Information Systems Group, University of Basel, Switzerland
[4] Information Systems Group, University of Duisburg-Essen, Germany
[5] Databases, Data Mining and Visualization Group, University of Constance, Germany
[6] Information & Software Engineering Group, Technical University of Vienna, Austria
[7] Institute for Information Systems, ETH Zurich, Switzerland
[8] Database and Information Systems Group, University of Constance, Germany

Contact Author: heiko.schuldt@unibas.ch

Abstract

DelosDLMS is a prototype of a next-generation Digital Library (DL) management system. It is realized by combining various specialized DL functionalities provided by partners of the DELOS network of excellence. Currently, DelosDLMS combines text and audio-visual searching, offers new information visualization and relevance feedback tools, provides novel interfaces, allows retrieved information to be annotated and processed, integrates and processes sensor data streams, and finally, from a systems engineering point of view, is easily configured and adapted while being reliable and scalable. The prototype is based on the OSIRIS/ISIS platform, a middleware environment developed by ETH Zürich and now being extended at the University of Basel.

Categories and Subject Descriptors

H.3 [Information Storage and Retrieval]: H.3.1 Content Analysis and Indexing, H.3.2 Information Storage, H.3.3 Information Search and Retrieval, H.3.4 Systems and Software, H.3.7 Digital Libraries

General Terms

Algorithms, Performance, Reliability

Keywords

Digital Library Management System, System Integration, Service-oriented Architecture

1 Introduction

The overall goal of the DelosDLMS is the implementation of a prototype of a next-generation digital library management system [Schek and Schuldt 2006]. This system combines text and audio-visual searching, offers personalized browsing using new information visualization and relevance feedback tools, provides novel interfaces, allows retrieved information to be annotated and processed, integrates and processes sensor data streams, and finally, from a systems engineering point of view, is easily configured and adapted while being reliable and scalable.

Previous work in the DELOS network has mainly focused on improving digital libraries (DLs) by developing independent, powerful and highly sophisticated prototype systems in cooperation of few partners within the different work packages. The goal of the DELOS integration activities is to integrate these prototype systems as building blocks into OSIRIS/ISIS, an existing middleware environment that was developed at ETH Zurich. The result of the integration – that is, the middleware infrastructure together with all the advanced DL functionality – constitutes the

DelosDLMS prototype. This paper reports on the integration results of the first phase which includes the integration of audio features (TU Vienna) and 3D features (Univ. of Firenze) into DelosDLMS, a new paper-based interface (iPaper from ETH Zürich), a SOM visualization of the feature space (Univ. Konstanz), and an annotation service (FAST, Univ. Padua) which have been added to DelosDLMS and the integration of DelosDLMS into the Daffodil interface.

The paper is organized as follows. Section 2 introduces the ISIS/OSIRIS middleware which is the basis for DelosDLMS. Section 3 presents in detail the services and DL functionality that has been added and architectural details on the integration. Section 4 concludes.

2 OSIRIS/ISIS Middleware

OSIRIS (Open Service Infrastructure for Reliable and Integrated process Support) is a platform that allows for combining different distributed services into processes [Schuler, Türker, Schek, Weber and Schuldt 2006]. The OSIRIS platform itself does not provide any application functionality but, by combining specialized application services, supports the definition and reliable execution of dedicated processes (this is also known as "programming-in-the-large"). When different specialized Digital Library application services are made available to the OSIRIS platform, users can define and run powerful Digital Library processes by making use of these services. This is realized by a set of generic (application-independent) services that include the registration of services and processes, interfaces for application development, an engine for decentralized execution of processes, and services for load balancing. OSIRIS processes themselves are wrapped by a service interface.

Therefore, a process can be invoked just like any other service (and used in other processes as well). Processes in OSIRIS contain two orders on their constituent services: a (partial) precedence order specifies regular execution while the precedence order is defined for failure handling purposes (alternative executions). Data flow between services of a process can be defined independently of control flow. Activities in a process are invocations of application services. Ideally, the transactional behaviour of each application service is known. This transactional behaviour includes information on compensation (how can the effects of a service execution be semantically undone; this is needed for compensation purposes in case a failure in a process execution exists) and on whether a failed service can be re-invoked (retriability).
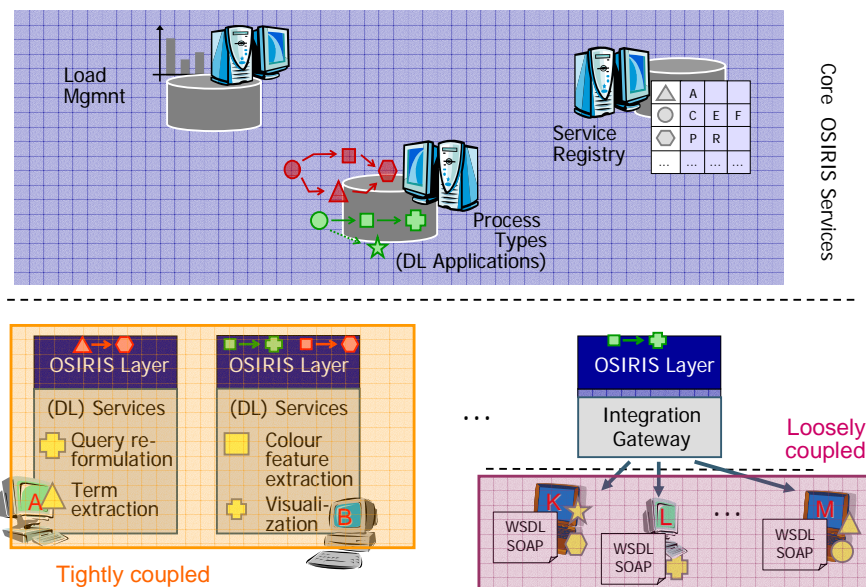

Figure 1: The OSIRIS Middleware at a Glance

OSIRIS distinguishes between system services and application services. System services are used internally for coordinating the execution of processes in a distributed way, without relying on a central execution engine/scheduler. For application services, OSIRIS further distinguishes between loosely coupled and tightly coupled services. Usually, a part of the distributed OSIRIS middleware (called OSIRIS layer) runs on each host providing application services. These application services are considered as tightly coupled since, in case information on their transactional properties like compensation or re-invocation is available, dedicated transactional guarantees can be provided at process level. Loosely coupled application services are those that have to be called remotely, without a local ORISIS layer available. The integration/invocation is done via WSDL for service description and SOAP for invocation. The system architecture of OSIRIS is depicted below.

In addition to transactional guarantees and reliability, OSIRIS focuses on scalability of process execution. The decentralized peer-to-peer approach for process execution in OSIRIS, which is realized by sophisticated replication mechanisms for control flow dependencies, avoids any single point of failure during process execution and provides a high degree of scalability. Peer-to-peer process execution also incorporates sophisticated load balancing in order to distribute process load among available, suitable peers.

Finally, OSIRIS is equipped with the O'GRAPE (OSIRIS GRAphical Process Editor) user interface for process definition. It allows for easy creation of process descriptions without programming skills. In addition, O'GRAPE supports the integration of existing application services by leveraging existing Web service standards like SOAP and WSDL.

The core of OSIRIS has been developed at ETH Zürich and has been further developed at UMIT, Austria. Currently, it is being extended (e.g., for processing continuous streams of data) at the University of Basel. There are two versions of OSIRIS: one is implemented in C++ and runs on Microsoft platforms, the other version is implemented in Java.

ISIS (Interactive SImilarity Search) is a prototype application for information retrieval in multimedia collections built at ETH Zürich [Mlivoncic, Schuler and Türker 2004]. It supports content-based retrieval of images, audio and video content, and the combination of any of these media types with text retrieval. Basically, ISIS consists of a set of pre-defined processes and several application services (like feature extraction, index management, index access, relevance feedback, etc.) that have been developed on the basis of the OSIRIS middleware. ISIS includes a sophisticated index structure (VA-file) for similarity searching, which is particularly well suited for high-dimensional vector spaces. It also provides basic support for relevance feedback and visualization.

With the DelosDLMS, existing ISIS services are significantly enriched by other specialized DL services that have been developed within the DELOS network. This is achieved by integrating these services into the OSIRIS infrastructure, thereby combining them with other ISIS and non-ISIS services into advanced, process-based DL applications.

3 Integration Activities

This chapter provides an overview of the achievements obtained in the task 1.8 integration activities. The starting point for the integration activities is illustrated in Figure 2. It includes several stand-alone systems (ISIS/OSIRIS, iPaper, Daffodil, FAST and SOM visualization) as well as specialized features for audio and 3D objects, respectively. All these components are presented in detail in the following subsections.
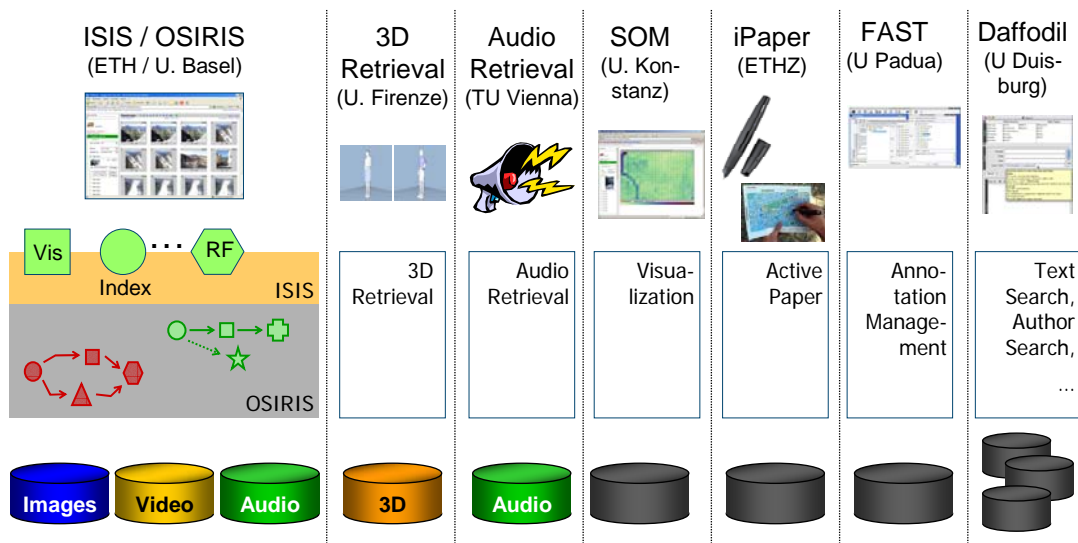
Figure 2: Overview of systems which have been integrated into DelosDLMS

3.1 Content-Based 3D Retrieval

Histograms of surface curvature have been used to support global description and retrieval of 3D objects. However, since histograms do not include any spatial information, they are liable to false positives. To overcome such problems, we used curvature correlograms as a model for representation and retrieval of 3D objects [Antini, Berretti, Del Bimbo and Pala 2005a], [Antini, Berretti, Del Bimbo and Pala 2005b]. Correlograms have been previously used with success for retrieval of images based on colour content [Huang , Kumar, Mitra, Zhu and Zabih 1999]. In particular, with respect to description based on histograms of local features, correlograms enable also encoding of information about the relative position of local features. Correlograms are used to encode information about curvature values and their localization on the object surface. For this peculiarity, description of 3D objects based on correlograms of curvature proves to be effective for the purpose of content based retrieval of 3D objects. In order to compute curvature correlograms of a 3D object, the principal curvatures are computed for every vertices of the mesh. Values of the mean curvature are then derived from principal curvature, and quantized into N classes of discrete values. For this purpose, the mean curvature value is processed through a stair-step function so that many neighbouring values are mapped to one output value.
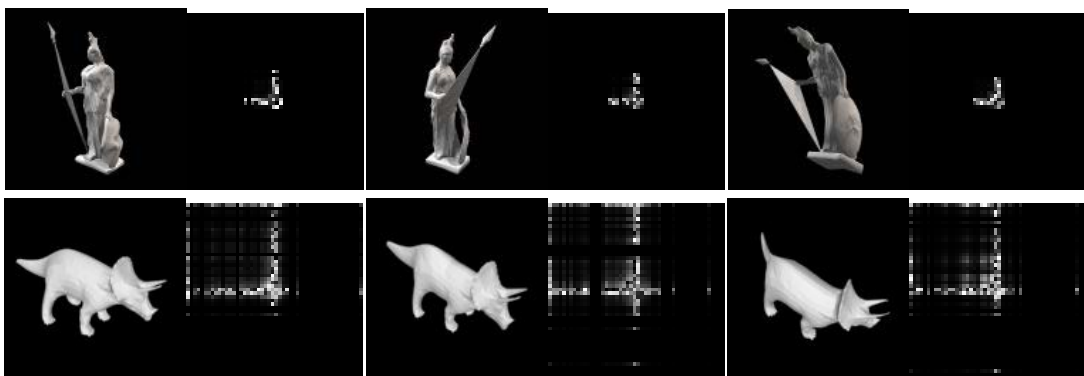

Figure 3. In (a) and (b) correlograms of three models, taken from two distinct categories, namely statue and dinosaur, are shown.

The correlogram of curvature is defined with respect to a predefined distance value δ. In particular, the curvature correlogram is defined as the probability that two vertices that are δ far away from each other, have curvature belonging to two classes of curvature c_i and c_j. In the current setting 50 classes of curvature are used, and δ = 5. Correlogram examples are shown in Figure 1.

The proposed retrieval approach has been applied to the models included in the Princeton Shape Benchmark database [Shilane, Min, Kazhdan and Funkhouser 2004]. This archive includes 1814 models categorized in a hierarchy of classes.

The integration of the curvature correlogram 3D retrieval approach developed at Uni Firenze, into the DelosDLMS, started in June 2006. During this activity, the curvature correlograms of the Princeton 3D-collection have been extracted and indexed with the efficient VAFile approach offered by DelosDLMS. A Web-service offering online feature extraction by Delos DLMS has been developed.

3.2 Audio Retrieval

TU Vienna has developed three different feature sets for content-based music retrieval. These feature sets for musical content description – Rhythm Patterns, Statistical Spectrum Descriptors and Rhythm Histograms – are integrated into DelosDLMS. The feature extraction algorithm is as follows:

In a pre-processing step the audio signal is converted to a mono signal and segmented into chunks of approximately 6 seconds. Typically, the first and last one or two segments are skipped and from the remaining segments every third one is processed.

For each segment the spectrogram of the audio is computed using the short time Fast Fourier Transform (STFT). The Bark scale, a perceptual scale which groups frequencies to critical bands according to perceptive pitch regions, is applied to the spectrogram, aggregating it to 24 frequency bands. The Bark scale spectrogram is then transformed into the decibel scale. Further psycho-acoustic transformations are applied: Computation of the Phon scale incorporates equal loudness curves, which account for the different perception of loudness at different frequencies. Subsequently, the values are transformed into the unit Sone. The Sone scale relates to the Phon scale in the way that a doubling on the Sone scale sounds to the human ear like a doubling of the loudness. This results in a Bark-scale Sonogram – a representation that reflects the specific loudness sensation of the human auditory system.

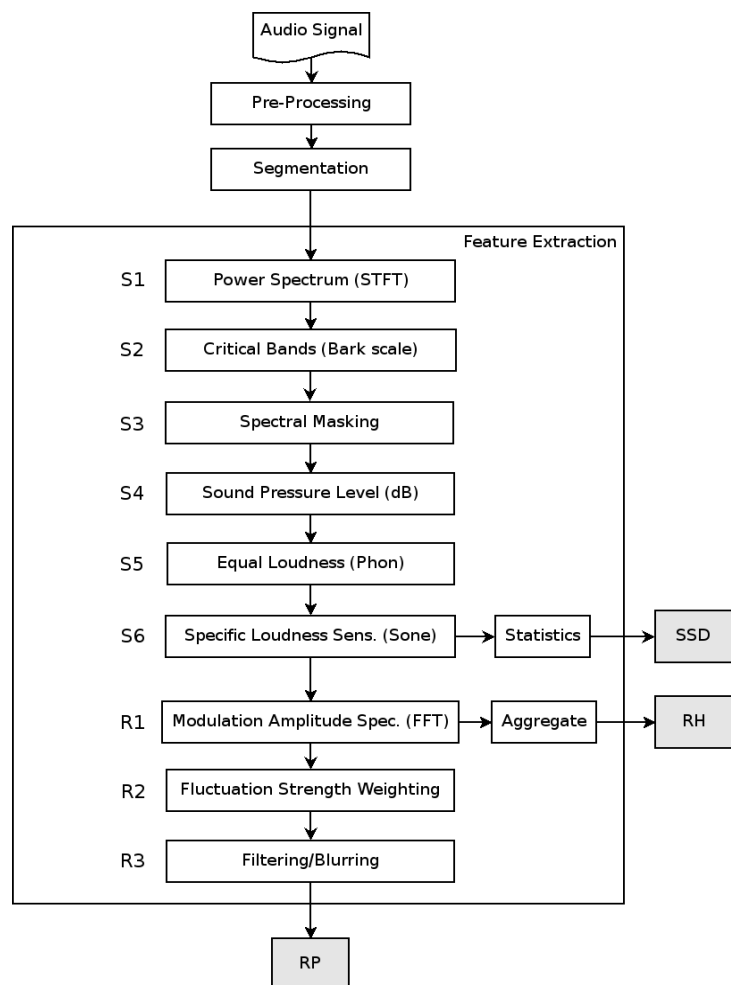From this representation of perceived loudness statistical measures (mean,

Figure 4: Audio Feature Extraction

median, variance, skewness, kurtosis, min and max) are computed per critical band, in order to describe fluctuations within the bands extensively. The se result is a Statistical Spectrum Descriptor.

In a further step, the varying energy on the critical bands of the Bark scale Sonogram is regarded as a modulation of the amplitude over time. Using a Fourier Transform, the spectrum of this modulation signal is retrieved. The result is a time-invariant signal that contains magnitudes of modulation per modulation frequency per critical band. This matrix represents a Rhythm Pattern, indicating occurrence of rhythm as vertical bars, but also describing smaller fluctuations on all frequency bands of the human auditory range. Subsequent to the Fourier Transform, modulation amplitudes are weighted according to a function of human sensation of modulation frequency, accentuating values around 4 Hz. The application of a gradient filter and Gaussian smoothing potentially improves similarity of Rhythm Patterns which is useful in classification and retrieval tasks.

A Rhythm Histogram is constructed by aggregating the critical bands of the Rhythm Pattern (before weighting and smoothing), resulting in a histogram of rhythmic energy for 60 modulation frequencies.

The feature vectors are computed for a piece of audio by taking the median of the descriptors of its segments.

3.3 SOM-Based Retrieval Support

A standard method to visualize the result set of a user query in a Digital Library is to display a sorted list of answer objects. Visualizing multiple object sets at once, as well as their structure and interrelationships between each other, possibly in different feature spaces, is feasible with projections from high-dimensional feature space to low-dimensional (2D) display space. Among other algorithms such as Principal Component Analysis and Multidimensional Scaling, the Self-Organizing Map (SOM) algorithm is a projection and vector quantization algorithm with interesting theoretical and practical properties, which is also highly suited for visualization. Previous work done at UNI-KN has focused on the application of the SOM algorithm for effective retrieval and visual analysis in multimedia databases. For DELOS, this work was leveraged in computing SOM projections for an image database described in different feature spaces. A web service was developed, which, based on the offline calculated maps, is able to render two basic SOM views, namely, U-matrices, and population density histograms (cf. the left-hand side image in Figure 5). Included is also a scheme for marking of query and answer objects on the map. The principal goal of the visualization is to support the user in analyzing and comparing the relationships between query and result sets, possibly under different feature representations. Furthermore, the visualization allows the user to effectively explore the contents and structure of a previously unknown data repository (visual analysis and scatter browsing).

Next steps in developing the SOM service could include the design and development of additional SOM views such as component planes, purity views, and such. Also, interaction functionality could be developed, where the user is allowed to selected interesting map regions and inspect the contents of such regions (cf. right-hand side image in Figure 5). Linking and combining SOM views over different feature representations of a given database, comparative visual exploration using the SOM could be supported.
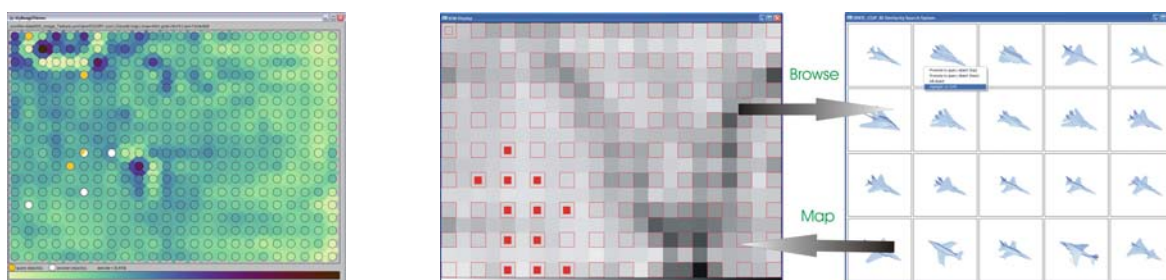
Figure 5. SOMs for certain DLMS database content were generated, and a first basic visualization service was implemented (left image). The SOM visualization technique is highly suited to support content-based retrieval and visual analysis in Digital Libraries (right image; previous work of UNI-KN).

3.5 Daffodil

The digital library system Daffodil is a front-end system for digital libraries, targeted at strategic support of users during the information search process. For searching, exploring, and managing digital library objects it provides user-customisable information seeking patterns over a federation of heterogeneous digital libraries. Searching with Daffodil makes a broad range of information sources easily accessible and enables quick access to a rich information space. Daffodil already provides a broad range of tools to help the user in query formulation, e. g. a thesaurus, a spell-checker, a classification browser, and a related-term service. For the user interface design, Daffodil implements a tool-based design paradigm, where objects resulting from the interaction with one tool can be dragged to other tools for further processing. This way, users can easily realize ad-hoc workflows involving different tools. Based on this metaphor, Daffodil implements the user-oriented integration of DL tools in the Delos DLMS, thus complementing the system-oriented integration performed by the OSIRIS middleware.

As a first reference implementation, the Daffodil system was extended by an image search tool connecting to the ISIS system described above. For this purpose, the Daffodil system was extended by two special functions:

[1] Query by Example: Given any picture link from the WWW, similar pictures are searched. The similarity feature can be chosen, e.g. colour histograms. A set of answers pictures is returned. The pictures can then be used for further searching.
[2] Query by Keyword: Given a specific keyword, the system is able to search for pictures, which are indexed with this keyword.

A feedback list is implemented in both functions. The user is able to specify relevant or non-relevant pictures in order to refine the answer set of searched pictures. The communication between OSIRIS and Daffodil is handled by HTTP and based on the processes given by the ISIS picture search. Through the integration within the Daffodil system, users can also apply other tools on queries and retrieved images, like maintaining a search history, storing queries and images within their personal libraries, sharing them with other users or adding annotations (see 3.7).

3.6 iPaper

Information managed by the DelosDLMS can be accessed by a variety of digital user interfaces. The goal of the iPaper extension for the DelosDLMS was to enhance existing paper documents with functionality offered by DelosDLMS thereby bridging the gap between paper documents and services offered by digital library management systems.

We developed an interactive printed museum guide as a physical user interface to the DelosDLMS. The system is based on a Bluetooth-enabled digital pen that can track user interactions within paper documents or any other surface covered with a special printed Anoto pattern consisting of tiny, almost invisible dots. Active paper areas can be defined within paper documents and linked to digital information or services. In the case of the interactive museum guide shown in Figure 1, existing printed text components and images, as well as some new interface components in the form of blue "paper buttons" have been associated with specific ISIS queries.
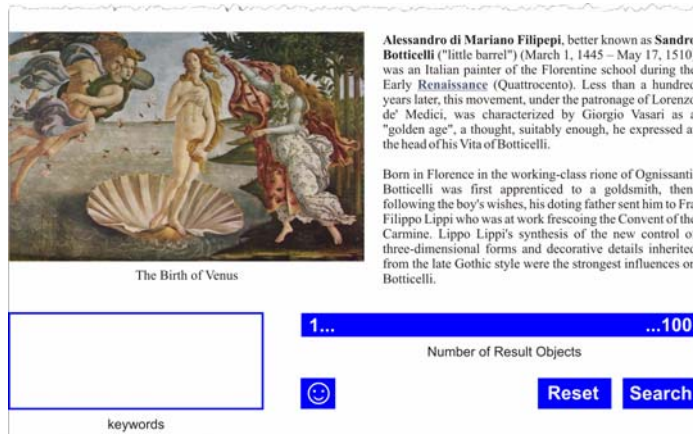

Figure 6: Paper-based ISIS query interface

A first query that can be formulated based on the paper interface is retrieving images based on a set of given keywords. A user can either select specific underlined words within the text, which is similar to selecting hyperlinks on the Web, or they can enter specific words in a special keyword input area. Any keywords entered in the keyword area are transformed to a text string by an Intelligent Character Recognition (ICR) process. An arbitrary number of keywords may be aggregated by selecting parts of the text or writing keywords. As soon as the user selects the 'Search' button with the digital pen, a query consisting of the set of keywords is sent to ISIS and a list of images matching these keywords is returned. Another type of query is based on finding image similarities. By touching images in the museum guide with the digital pen, a user can add them to the set of images to be used in a similarity search. After selecting the 'Search' button, a query with the set of selected images as parameter is sent to ISIS and a list of best matching images is returned. Note that the keyword and image-based searches may be combined and queries containing keywords as well as images can be formulated. Finally, there are some paper buttons to configure specific query parameters such as the preferred size of the result set (from 1 up to 100 images) or the ISIS face recognition functionality. Furthermore, at any time it is possible to reset existing query parameters by selecting the 'Reset' area with the digital pen.

The iPaper infrastructure [Norrie, Signer and Weibel 2006], [Norrie, Palinginis and Signer 2005] used to implement the paper-based interface for DelosDLMS was developed by the Global Information Systems (GlobIS) research group at ETH Zurich. It is based on a general cross-media information management server (iServer) [Norrie and Signer 2005] enabling the integration of arbitrary digital and physical resources.

3.7 FAST Annotation

FAST [Agosti and Ferro 2005a] is a flexible system designed to support both various architectural paradigms and a wide range of different DLMSs. In order to achieve the desired flexibly: 1) FAST is a stand-alone system, i.e., it is not part of any particular DLMS; 2) the core functionalities of the annotation service are separated from the functionalities needed to integrate it into different DLMSs. The flexibility of FAST and its independence from any particular DLMS are key features to provide users with a uniform way of interaction with annotation functionalities. In this way, users do not need to change their annotative practices only because they work with different DLMSs.

From an architectural point of view, FAST adopts a three-layers architecture – the data, application and interface logic layers – and is designed at a high level of abstraction in terms of abstract Application Program Interfaces (APIs) using an object-oriented approach.

Within FAST, annotations are composite multimedia objects, which can annotate multiple part of a given digital object and can relate this annotated digital object to various other digital objects, if needed. Furthermore, once it has been created, an annotation is considered as a first class digital object, so that it can be annotated too. In this way, users can create not only sets of annotations concerning a digital object, but also threads of annotations, i.e., annotations which reply to one another. These threads of annotations are the basis for actively involving users with the system and for enabling collaboration

From a functional point of view, FAST offers annotation management functionalities, such as creation, access, and so on. Furthermore it supports collaboration among user by introducing scopes of annotation and groups of users: annotations can be private, shared or public; public; if an annotation is shared, different groups of users can share it with different permissions, e.g., one group can only read the annotation while another can also modify it.

Finally, FAST offers advanced search functionalities based on annotations by exploiting annotations as a useful context in order to search and retrieve relevant documents for a user query [Agosti and Ferro 2005b], [Agosti and Ferro 2006]. The 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.

3.8 Summary of Integration Activities

All integration activities of the first phase are summarized in Figure 7. Both audio and 3D features have been added to the VA-file index of DelosDLMS. Since the FAST annotation management and the SOM visualization are available as web services, they have been loosely integrated into DelosDLMS and OSIRIS processes have been extended accordingly in order to make use of these new services. The active paper front-end creates a query which is sent and executed in DelosDLMS. Similarly, the Daffodil client generates a query which is sent to DelosDLMS. The result objects are displayed in the Daffodil client.
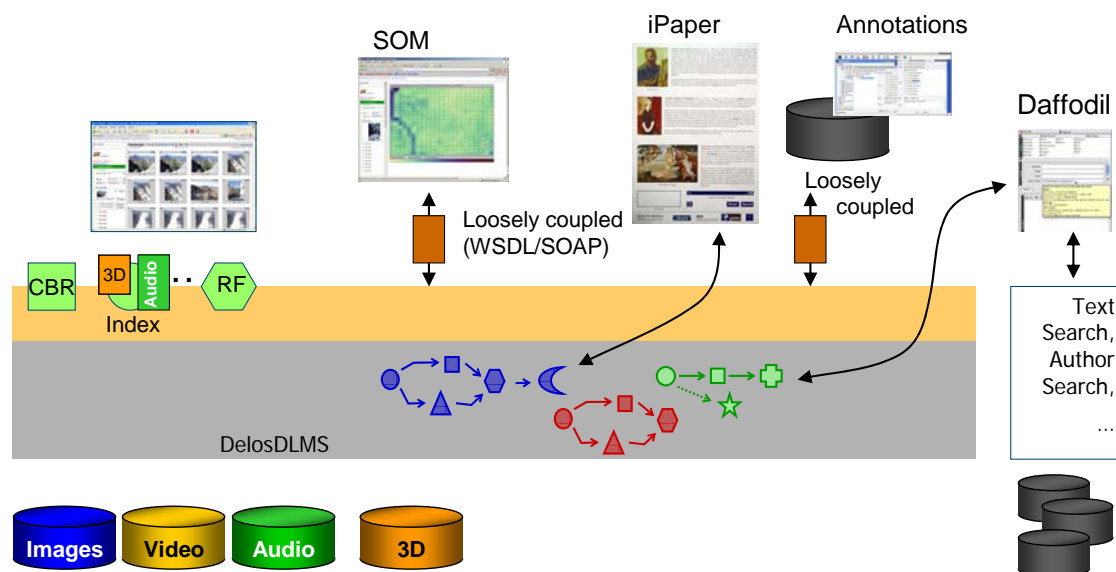

Figure 7: Status of DelosDLMS as of January 2007

4 Summary and Outlook

The DelosDLMS prototype development will continue in the final year of the DELOS network. First, the services which are already integrated in a loosely-coupled way will be integrated more tightly. Second, some functionality like audio features and 3D features has been integrated off-line, by loading the feature vectors into the DelosDLMS VA-file index. It is planned to make feature extractors available online and to create OSIRIS processes that allow for the upload and indexing of new objects and/or for searching with objects that are not yet indexed. Third, new functionality will be added. This includes personalisation, term extraction, query support for XML documents, semantic video services, video annotation, and video retrieval, P2P search, natural language and speech interfaces and further visualizations.

References

Agosti M. and Ferro N. 2005a. 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.* Lecture Notes in Computer Science (LNCS) 3664: 147–166

Agosti M. and Ferro N. 2005b. Annotations as context for searching documents. In F. Crestani and I. Ruthven (eds.), *Information Context: Nature, Impact, and Role. Proceedings of the 5th International Conference on Conceptions of Library and Information Sciences.* Lecture Notes in Computer Science 3507. Heidelberg: Springer. 155–170.

Agosti M. and Ferro N. 2006. Search Strategies for Finding Annotations and Annotated Documents: the FAST Service. In H. Legind Larsen, G. Pasi, D. Ortiz-Arroyo, T. Andreasen, and H. Christiansen (eds.). *Proceedings of the 7th International Conference on Flexible Query Answering Systems (FQAS 2006).* Lecture Notes in Artificial Intelligence (LNAI) 4027. Heidelberg: Springer. 270–281.

Antini G., Berretti S., Del Bimbo A. and Pala P. 2005a. Curvature Correlograms for Content Based Retrieval of 3D Objects. In F.Roli and S.Vitulano (eds.). *Proceedings of Int. Conf. on Image Analysis and Processing (ICIAP'05).*Cagliari (Italy). Lecture Notes in Computer Science 3617. 859-866.

Antini G., Berretti S., Del Bimbo A. and Pala P. 2005. Retrieval of 3D Objects using Curvature Correlograms. In Proceedings of Int. Conf. on Multimedia and Expo (ICME'05). Amsterdam (The Netherlands). July 2005. 735-738.

Huang J., Kumar R., Mitra M., Zhu W.-J. and Zabih R. 1999. Spatial Color Indexing and Application. In International Journal of Computer Vision 35: 245-268.

Kriewel S., Klas C.P., Schaefer A. and Fuhr N. 2004. Daffodil - Strategic Support for User-Oriented Access to Heterogeneous Digital Libraries.In D-Lib Magazine 6(10).

Mlivoncic M., Schuler C. and Türker C. 2004. Hyperdatabase Infrastructure for Management and Search of Multimedia Collections. In DELOS Workshop: DL Architectures : 25–36.

Norrie M.C., Palinginis A. and Signer B. 2005. Content Publishing Framework for Interactive Paper Documents. In Proceedings of DocEng 2005, ACM Symposium on Document Engineering. Bristol (UK).

Norrie M.C. and Signer B. 2005. Information Server for Highly-Connected Cross-Media Publishing. In Information Systems 30(7).

Norrie M.C., Signer B. and Weibel N. 2006. General Framework for the Rapid Development of Interactive Paper Applications. 1st International Workshop on Collaborating over Paper and Digital Documents.

Schaefer A., Jordan M., Klas C.P., Fuhr N. 2005. Active Support For Query Formulation in Virtual Digital Libraries: A case study with DAFFODIL. In Proceedings ECDL.

Schek, H.-J. and Schuldt, H. 2006. DelosDLMS - Infrastructure for the Next Generation of Digital Library Management Systems. In: ERCIM News No. 66, Special Issue on European Digital Library, pages 22-24, July 2006.

Schuler C., Türker C., Schek H.-J., Weber R. and Schuldt H. 2006. Scalable Peer-to-Peer Process Management. In International Journal of Business Process Integration and Management (IJBPIM) 1(2) : 129-142.

Shilane P., Min P., Kazhdan M. and Funkhouser T. 2004. The Princeton Shape Benchmark. *Shape Modeling International.* Genova (Italy).