Int J Digit Libr (2008) 9:101–114
DOI 10.1007/s00799-008-0044-y

REGULAR PAPER

DelosDLMS

From the DELOS vision to the implementation of a future digital library management system

Yannis Ioannidis · Diego Milano · Hans-Jörg Schek · Heiko Schuldt

Published online: 14 October 2008

Abstract DelosDLMS is a novel digital library management system (DLMS) that has been developed as an integration effort within the DELOS Network of Excellence, a European Commission initiative funded under its fifth and sixth framework programs. In this paper, we describe DelosDLMS that takes into account the recommendations of several activities that were initiated by DELOS including the DELOS vision for digital libraries (DLs). A key aspect of DelosDLMS is its novel generic infrastructure that allows the generation of digital library systems out of a set of basic system services and DL services in a modular and extensible way. DL services like feature extraction, visualization, intelligent browsing, media-type-specific indexing, support for multilinguality, relevance feedback and many others can easily be incorporated or replaced. A further key aspect of DelosDLMS is its robustness against failures and its scalability for large collections and many parallel user requests. We discuss the current status of an effort to build DelosDLMS, a Digital Library Management System that integrates in various ways several components developed by DELOS members and showcases a great variety of functionality that is outlined as part of the DELOS vision.

Keywords Digital libraries (DLs) · DLMS · Programming in the large

Y. Ioannidis
Department of Informatics,
National and Kapodistrian University of Athens, Athens, Greece
e-mail: yannis@di.uoa.gr

D. Milano · H. Schuldt (✉)
Database and Information Systems Group,
University of Basel, Basel, Switzerland
e-mail: heiko.schuldt@unibas.ch

D. Milano
e-mail: diego.milano@unibas.ch

H.-J. Schek
Emeritus of ETH Zurich, Zurich, Switzerland
e-mail: schek@inf.ethz.ch

1 Introduction

Digital libraries (DLs) represent the meeting point of a large number of technical areas within the field of informatics, i.e., data management, information retrieval, document management, information systems, the web, image processing, artificial intelligence, human–computer interaction, mass-storage systems, and others. Moreover, DLs draw upon several other disciplines and fields beyond informatics, such as library sciences, museum sciences, archives, sociology, psychology, etc. DLs first appeared as a concept in the early 1990s and grew up to become a discipline in its own right through many individual projects that mostly focused on bridging some of the gaps between the constituent fields, understanding what 'digital library functionality' is supposed to be, and integrating solutions from each field into systems that support such functionality.

The results of most of these efforts are systems that manage particular DLs (collections of content of particular theme(s), to be exact) serving the needs of particular users. Currently, researchers, educators, students, and members of other communities continuously search DLs for information, as part of their daily routines, decision making, or entertainment. Whether they do use the term 'digital library' (such as the "Perseus Digital Library" [36], the "ACM Digital Library" [1], or even the "Children's Digital Library" [17]) or not (such as "The Online Books Page" [37]), there are many systems that employ relevant technology. Their functionality

has improved the quality of their users' activities' results dramatically, facilitating better information provision and more advanced research. Nevertheless, the current state of affairs is still severely restricted as almost all DLs developed so far share the following characteristics:

Content-centric systems: Motivation to build DLs has been based on the need to organize and provide access to particular collections of data and information, more or less ignoring any particular usage requirements.

Storage-centric functionality: Static storage and retrieval of information has been assumed to be the main role of a digital library.

Isolated and repeated efforts: Each digital library system has been built from scratch, independently of any other development, and reusing nothing that is already available, hence repeating the work associated with several system components that are common.

Environment-specific systems: As a result of the previous characteristic, every digital library system has been tailored to the particular needs and characteristics of the target environment with little provision for generalization.

Isolated systems: Even after development, every digital library system has mostly operated in isolation, without any interactions with other similar systems. Such fragmentation has limited the opportunities for large-scale analysis and global-scale information availability as well.

Application restrictions: Past digital library development has mostly focused on material that is traditionally found in Libraries, and in fact, mostly related to cultural heritage.

Hence, despite the undisputed advantages that current digital library systems offer compared to the pre-1990s era, the above restrictions limit the role that DLs can play in knowledge societies, which will serve as important educational nuclei in the future.

Almost 10 years ago, the DELOS Network of Excellence on digital libraries[1] initiated a long journey towards filling the gap between current digital library practice and the needs of modern information provision. It has played a significant role in the formation of an active DLs research community, the formulation of a vision for the future of the field, and in fostering collaborative research in the direction of this vision. Its goal was to foster the development of technology that will eventually overcome all six of the above restrictions of today's systems and empower everyone to further advance their knowledge, profession, and role in society.

In this paper, we outline the DELOS vision for DLs, and we focus on technological factors that must be considered as prerequisites to fulfilling such vision. We then describe DelosDLMS, a digital library management system (DLMS) developed through a common effort by DELOS members, showing how it relates to such requirements and satisfies them. The main contribution of this paper consists in the presentation of the novel infrastructure of DelosDLMS that allows the generation of various different DLs and DL applications. The system has been successfully implemented and demonstrated and shows that the visionary principles can be turned into a powerful new and real system which shows properties and functionality that is not available in any other known system so far.

The remainder of this paper is structured as follows. Section 2 presents the DELOS vision for DLs. Section 3 identifies the technical requirements stemming from the DELOS vision, introduces DelosDLMS and shows how it allows to meet these requirements. Sections 4 and 5 describe in detail the main architectural components of DelosDLMS. Section 6 shows some concrete examples of how DL applications are built on the basis of DelosDLMS. Section 7 concludes.

2 The DELOS vision

Ultimately, DELOS envisions systems with no logical conceptual, physical, temporal, or personal borders or barriers on information. The DELOS vision is that DLs will become the universal knowledge repositories and communication conduits for the future, common vehicles by which everyone will access, analyze, evaluate, enhance, and exchange all forms of information [21]. They will be indispensable tools in the daily personal and professional lives of people. They will be accessible at any time and from anywhere, and will offer a friendly, multi-modal, efficient, and effective interaction and exploration environment [18].

Clearly, any effort towards this vision requires significant change in the digital library development strategies of the past, with respect to functionality, operational environment, and other aspects. Contrary to current practice, the vision requires that digital library systems have the following characteristics:

Person-centric efforts: Humans are at the center of DLs and all efforts to develop them should be initiated and motivated by needs to provide interesting and/or novel experiences to users. Furthermore, digital library systems should synthesize all information that is available about each person in a cohesive whole, so that they may offer personalized treatment to individuals or classes of individuals based on their profiles.

Communication-centric and collaboration-centric functionality: The main role of DLs must be to facilitate interaction of scientists, researchers, or the general public on themes that are pertinent to the information stored. Storage of this information and access to it is only a small part of such functionality.

Generic technology systems: For economy of scale, and in order to facilitate reusability and extensibility, generic

[1] http://www.delos.info.

DLMSs should be developed that capture all common management aspects of DLs. Supporting any further, environment-specific needs on content manipulation or user interfaces should be developed in a customized fashion on top of DLMSs.

Maximum-reuse efforts: Given the existence of industrial-strength DLMSs, every development effort should depend on them, avoiding much mundane work that is currently necessary, and should only focus on the specialized parts.

Globally distributed systems: DLs should be managed by widely distributed systems, through which information sources across the world get interconnected to exchange and integrate their contents.

Universality of information and application: DLs should be put in the service of "all" applications and should comprehensively manage "all" forms of content, from data to information to knowledge.

The DELOS Network of Excellence aimed at advancing the state of the art in the field so that a first version of the vision could become reality by the end of the decade. It focused on key instances of the above features and attempted to deliver novel technology and robust prototype demonstrators that will be characterized by them.

3 From the DELOS vision to DelosDLMS

From the DELOS vision presented in the previous section, both long term goals and short to mid term goals can be derived. The long term aspects include the role of DLs in society and their relationship to DL users. This needs to be pursued by the digital library community at large, including many contributing disciplines participating to the creation, operation and maintenance of DLs.

On the other hand, some of these challenges are clearly technological in nature, and it is mainly the computer science research community that has the role to provide solutions for them. The availability of such solutions is both a goal per itself, and an enabler for a further diffusion of the ideas behind the DELOS vision.

The paper focuses on the latter challenges, i.e., the technical requirements stemming from the overall DELOS vision, and shows how DelosDLMS, a prototype system for future DLs, implements this technical vision to a great extent.

3.1 Inside the DELOS vision: technical requirements

As described before, current digital library systems are often *content- and storage-centric* (focusing mostly on storage and access to information, often managed under a single form—e.g., textual documents) and *isolated* (not distributed, not interconnected and not capable of allowing the integration of content from different sources). Future DLs should instead be characterized by:

- The ability to *jointly manage different media types* (not only textual documents) and associated metadata; moreover future DLs should be extensible in terms of being able to add support for managing and accessing novel media types.
- Support for *communication and cooperation* for users that aim at cooperating, exchanging, sharing and/or jointly processing content, metadata, annotations, etc.
- *Distribution*. Future DLs are not isolated systems. Rather, they are supposed to cross organizational and institutional boundaries in a networked world. This means they are globally distributed systems with users that are able to access in an effective and efficient way content spread over multiple information sources across the Internet.

A DLMS [14] is a software system that captures all common management aspects of DLs, and supports the easy development of digital library systems (DLS)—pretty much in the same way a database management system (DBMS) facilitates building and managing databases and database applications. Ideally, a DLMS should make available to the developer an environment to create a DLS and at least a basic set of DL functionality. Any further application-specific requirements on content manipulation or user interfaces should be developed in a customized fashion on top of DLMSs.

The minimal functionality that a DLMS should offer in order to enable future DLs can be derived from the requirements given above. They include versatile support for content management, including customizable facilities for indexing, searching, annotating, and processing content in various ways.

The advantages and key aspects of building digital library systems based on DLMSs over the current practice to development essentially include *economy of scale*, as relying on a DLMS layer will take away from the developer much infrastructure work that is inevitably necessary, allowing her to focus only on the specialized parts with only little effort needed for extending or re-customizing the system. A further advantage is *richer functionality*, as the DLMS itself will incorporate and facilitate the reuse of many existing DL functionalities which otherwise would be hard to develop from scratch. Standardized interfaces will also foster the production of reusable components offering generic DL functionality, which can then be incorporated into a DLS at low cost.

From a systems point of view, the following characteristics are required for a DLMS in order to be compliant with the overall DELOS vision:

Extensibility: a key feature of DLMSs should be support for developing new functionality that is easily integrated with that already present.

Customizability: DLMSs have to be generic in a sense that they can be used as basis for the development of concrete DLSs for specific environments.

Reuse support: generic and specialized functionality developed on top of a DLMS should be easily ported to other DL Systems or reused to build new applications.

Robustness and scalability: Digital library systems running on top of a DLMS should be robust and should scale well to satisfy the growth of the DL in terms of content, of new features added, and of dimension of the user community targeted.

3.2 DelosDLMS: enabling the DELOS vision

DelosDLMS is a next-generation DLMS [30,5] which combines a novel generic distributed DL infrastructure, a set of basic system services, and a set of highly sophisticated DL services—all needed to build powerful digital library systems.

The philosophy behind DelosDLMS is to allow fast, easy development and flexible management of complex DL applications by relying on a "programming in the large" approach. In this approach, applications consist of processes, that describe the interaction among existing services. A middleware layer offers an execution environment for the processes. In other words, DL applications are built out of independently created building blocks, with a middleware acting as a glue between them.

The architecture of the system is organized over two logical layers: (1) a middleware layer which follows the paradigm of a service-oriented architecture (SOA) and offers the functionality to combine independent services into applications, and (2) an extensible library of DL services and interfaces that can be used as building blocks for DL applications.

This approach meets many of the requirements highlighted in the previous section. It allows fast development of DL applications, as building blocks covering the main functionality are already available in the system, and composing them amounts to design a process which incorporates them. It is highly modular, as independent functionality is provided by independent services. This favours reuse, as the same service can be used in different applications. Furthermore, extensibility is provided at two levels. The DLMS itself can be extended with new functionality by adding new services to its library. The only requirement is that such services are compliant with well-established standards from the service-oriented computing world (e.g., that a service is available as SOAP web service and described via WSDL). Process-based applications developed with this approach can also be made available as service (i.e., be subject to re-use in other applications). In addition, they can be easily extended, for instance by modifying the related process to incorporate a new service, without altering the rest of the application. The approach is also intrinsically suited for use in a networked environment where different services within a process-based DL application are made available by different, distributed providers.

DelosDLMS is based on a middleware layer which is provided by OSIRIS (Open Service Infrastructure for Reliable and Integrated process Support) [33,32], a system initially developed at ETH Zurich and then expanded and maintained at the University of Basel. OSIRIS fully supports the programming in the large approach. Services can be composed into processes. Processes, in turn, are wrapped by a service interface: they can be started and invoked like any other service and thus seamlessly incorporated into other processes.

Furthermore, application development in OSIRIS is strongly facilitated. Processes can be designed without being aware of many details regarding services (except for their interfaces), including their physical location and availability. Process invocation abstracts from specific service instances and frees the user or application from having to deal directly with failure handling. Finally, a graphical interface allows process definition through a boxes and arrows approach.

Additional benefits of using OSIRIS come from its distinctive features, which are not found in other SOA middleware. In OSIRIS, the execution of processes is completely distributed in a peer-to-peer (P2P) fashion. This provides a highly reliable and scalable execution environment for the resulting applications, fulfilling another fundamental requirements of future DL applications.

The support of the OSIRIS middleware for the development of DL applications is presented in the next section; more details can be found in [12].

In addition to the generic middleware, DelosDLMS features a rich set of independently developed DL services which can be flexibly combined. Using these services and the facilities provided by the OSIRIS platform, it is possible to define and run powerful digital library applications offering, for instance, combined text and audio-visual searching, personalized browsing using new information visualization and relevance feedback tools, novel interfaces, facilities to annotate retrieved information and manipulate annotations. Figure 1 illustrates the logical parts of DelosDLMS.

4 The core of DelosDLMS: OSIRIS

As stated in the previous section, DelosDLMS is based on OSIRIS, a middleware for distributed and decentralized P2P process execution. OSIRIS is the "glue" that allows combining distributed DL services into full fledged DL applications.

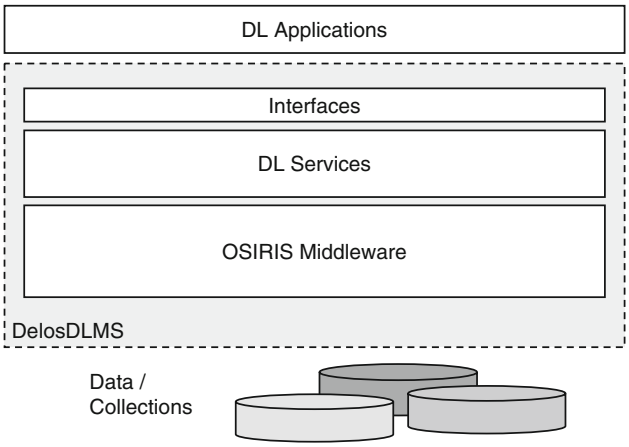

Fig. 1 Logical Components of DelosDLMS

Compared to other systems for service composition and process execution, OSIRIS has some key aspects that make it highly reliable and scalable.

Processes in OSIRIS follow the model of transactional processes [31]. They contain two orders on their constituent services: a (partial) precedence order for regular execution and a precedence order for failure handling purposes. The latter specifies alternative executions i.e., services and the order of their execution order which will be used when the regular execution leads to a failure situation—similar to exception handlers in programming languages. Data flow between services of a process can be defined independently of control flow. Activities in a process are invocations of (DL) 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).

In addition to transactional guarantees and reliability, OSIRIS focuses on scalability of process execution. The decentralized P2P 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. P2P process execution also incorporates sophisticated load balancing mechanisms in order to distribute process load among available, suitable peers.

All these features are supported by a set of internal *System Services*, used for communication and to coordinate distributed process execution and load management. Some of these services constitute an *OSIRIS local layer*, which is run at any node in an OSIRIS managed network.

The integration of new DL services into OSIRIS can be done by deploying them over an OSIRIS layer running at some node in a network of peers. Services deployed in this way are called tightly coupled services. Tightly coupled services are under full control of the OSIRIS layer. In case information on their transactional properties like compensation or re-invocation is available, dedicated transactional guarantees for processes can be provided. Loosely-coupled DL services are those that have to be called remotely, without a local OSIRIS layer available. The integration/invocation is done via WSDL for service description and SOAP for invocation.

4.1 System services and their functionality

System services provide general support for the execution of distributed and decentralized process-based (DL) applications. Among the system services, the global repositories are important since they store all necessary information for process execution. The OSIRIS layer of each node offers a replication service with is in charge of replicating sufficient information in order to allow for decentralized and decoupled process execution. Additional system services support tasks like process management, replication, messaging, routing, and load balancing.

Global repository services. The *Process Repository* holds the global definitions of all processes types (DL applications). Process definitions are decomposed into pairs of subsequent activities within the control flow, called execution units. Execution units are tuples containing process ID, ID of current activity, ID of subsequent activity/activities in control flow, and event type. On termination of the current activity, the OSIRIS layer produces an event to indicate the status of current activity execution. For example, if the current activity has not terminated gracefully, an error event is produced. The event type allows for different control flows for correct and failed execution. Additionally, one activity may have multiple direct successors in the control flow for the same event type, which corresponds to a split in the process execution. Only locally needed execution units are replicated. In other words, only execution units with current activities a provider is able to perform are replicated from this repository.

The *Service Registry* records a list of all available (tightly coupled) services, offered by all providers in the OSIRIS network. Providers in the OSIRIS network only need replicated information about services which are subsequent activities of execution units they are involved in.

The *Load Repository* is storing information of the current load situation of providers in the OSIRIS network. This information is replicated in a similar way as information from the subscription repository. Additionally, freshness parameters on subscriptions are used to avoid unnecessary propagation of minor load updates.

In addition, the *UDDI Registry* is in charge of keeping a registry of loosely coupled services together with their

WSDL descriptions, since loosely coupled services do not register themselves at the subscription repository. This registry is necessary to keep the OSIRIS infrastructure informed about available loosely coupled services.

All these repositories do not require a centralized service implementation. If the number of nodes in the OSIRIS network is increasing, additional repositories/services of same kind can reduce the load (i.e., by applying horizontal partitioning).

OSIRIS' decentralized and distributed approach to process execution is illustrated in Fig. 2. Different service types are illustrated with different shapes. The precedence order is represented by directed edges between the services of a process (i.e., the process on the left upper corner, whose execution is illustrated in the center of the figure, consists of three services which are invoked sequentially). The OSIRIS layer is available on top of all DL service providers and allows them to make available their services to OSIRIS processes. In the center, some of the OSIRIS metadata repositories are displayed, such as the *service registry* which keeps track of all the services available in the system, the *process programs registry* which lists all the process-based applications, and the *load repository* which manages approximations of the load of all service providers and which helps in balancing the load across several providers of the same service. All these repositories are not needed for the execution of a process. Rather, information out of these repositories is replicated to the local OSIRIS layer (dotted lines). Figure 2 also shows how a process description is divided into execution units (pairs of subsequent services) and how these execution units are replicated to the local OSIRIS layers of nodes providing participating tightly coupled services. Also necessary load and available service information is replicated from load and subscription repositories. Finally, after replication, enough process and service meta information is locally available to allow for P2P process execution. In particular, when a process is instantiated and executed, the local OSIRIS layers can decide on their own, based on locally replicated information, where to route a request to (solid lines between OSIRIS layers). This makes sure that process execution takes place in a decentralized and distributed way and guarantees a high degree of scalability.

4.2 Developing DL applications with O'Grape

OSIRIS comes with an integrated graphical administration tool, called O'Grape (OSIRIS Graphical Process Editor). O'Grape supports the design of new processes through a boxes and arrows approach. The intuitive interface of O'Grape makes the process of building applications in OSIRIS easy and fast. A screen shot of the interface of O'Grape in which a DL process is displayed is shown in Fig. 3. It shows how the paradigm of programming in the large can be supported in a graphical way and also shows how additional information on data flow and transactional semantics (lower and left hand side of the screen shot) can be specified.

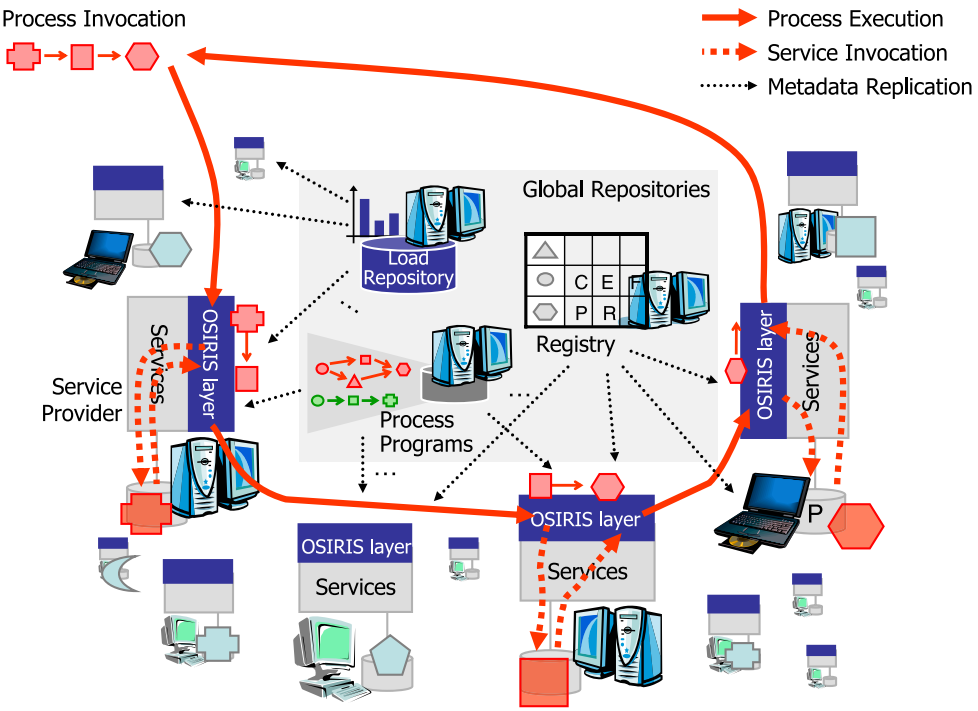

Fig. 2 OSIRIS Architecture and distributed execution of OSIRIS processes

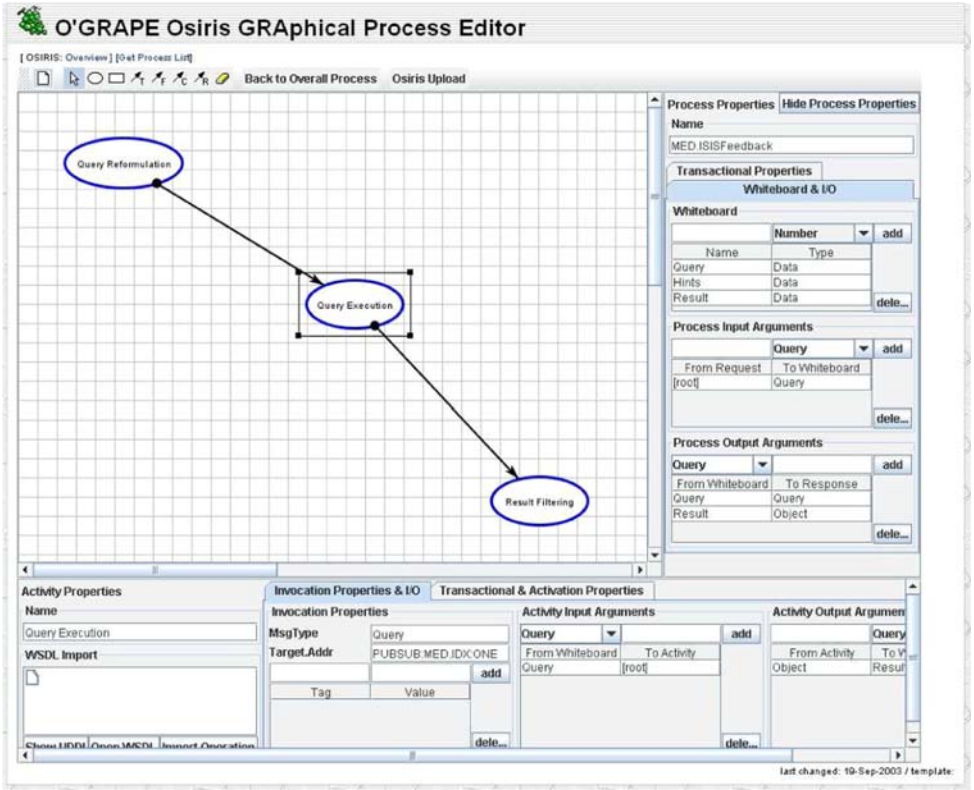

Fig. 3 O'Grape process design interface

OSIRIS exists in two versions, a C++ version for the Microsoft Windows platform, whose development began at ETH Zurich, and a cross-platform Java version, developed at the University of Basel. The O'Grape tool is also written in Java.

5 The DL services of DelosDLMS

On top of OSIRIS, DelosDLMS features a rich library of DL services. These range from very basic and generic services, like support for multimedia content storage, indexing and retrieval, to very specialized ones, like thesaurus-based query expansion support.

All these services define the DL functionality that DelosDLMS provides to its users "out of the box". Complete and sophisticated DL applications can be built starting from this collection of basic services. Concrete examples of how this is done are given in the next section. However, due to the modularity of the DelosDLMS approach, the library is extensible. It is possible to add new functionality by implementing new services (either as OSIRIS native services or as SOAP web services), that can then be freely composed with existing ones to create new DL applications.

When DelosDLMS started, the service library consisted originally of services belonging to ISIS (interactive similarity search), a system for multimedia search, tightly linked to OSIRIS (ISIS has been developed in parallel to OSIRIS at ETH Zurich and later extended at the University of Basel). Notable features of ISIS are the capability of mixing textual (metadata-based) and content-based queries, a high-performance indexing and retrieval service based on the VA file data structure [40] enabling multi-feature indexing over multimedia collections, and relevance feedback tools for iterative refinement of queries. Over time, the library has grown to include several DL services independently developed by DELOS partners. In what follows, we describe the current status of the DelosDLMS service library.

The functionality made available by the services in DelosDLMS covers many different needs that may arise in the construction of sophisticated DL applications. Many of the services are themselves complex systems, and our goal here is not to describe them in full detail but to give a general overview of the capabilities of the system. To help the reader understand the role of the various services, we adopt a classification based on their main functionality. The classes of services are shown at a glance in Fig. 4, and described below.

Digital library applications usually have to cope with managing, indexing and efficiently querying large collections of digital documents of various types, from textual documents to images and audio to collections of compound objects. In order to offer a generic support for building DL applications, DelosDLMS includes various generic multimedia *Content and Collection Management Services*, and *Search*

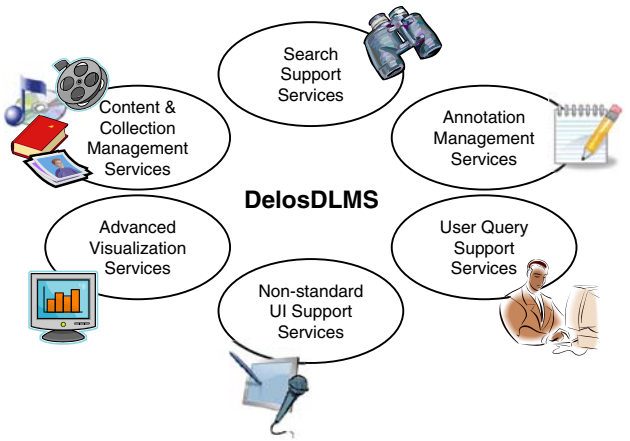

Fig. 4 Classes of DL services in DelosDLMS

Support Services to enable efficient metadata and content-based indexing and retrieval of content in the collections. Beside storing content, it is important to provide users with tools that increase their productivity and enhance their querying and browsing experience while interacting with DL applications. For this reason, DelosDLMS features *annotation management services*, that provide extensive support for handling annotation of content, including annotation sharing to support collaborative environments and querying annotations together with other metadata. Furthermore, a set of *User Query Support Services* improves the user experience during searches, allowing query expansion based on controlled vocabularies, incremental refinement of query results, and personalized interaction with the system. *Advanced Visualization Services* offer a powerful alternative to the classical table or list based display of query results and also allow to visualize, browse and analyze in an effective way entire collections. Finally *Non-standard UI Support Services* give the possibility to interact with DL applications through non-traditional input–output devices.

Given the open and extensible nature of the DelosDLMS service library, the classes listed below only describe the current state of DelosDLMS functionality, and their number might grow in the future. In the remainder of this section, we describe in more detail the services of DelosDLMS, following the classification given above.

5.1 Content and collection management services

Every object contained in a digital library is assigned a unique identifer, the OID, within the collection. The *Meta Database component* maintains the association of OIDs with objects. Each object is associated with a object type (OType), which can be one of the following types: image, audio, video, video sequence, or text. Every object can be associated with several locations, e.g., the original URL from which the object has been downloaded and the filename of the local copy, or the address of a thumbnail image. It is also possible to store and efficiently retrieve arbitrary additional properties such as size in bytes of the file or whether the object is copyrighted. The Meta Database service also maintains the structure of a collection, that is, the objects can be assigned to a taxonomy of categories.

The *Storage service* is able to deliver the file content of the object and monitor changes in a directory of the file system. In case a change is detected, a process is started in OSIRIS. For instance, if a new file is inserted, the storage service triggers the execution by sending a message to the OSIRIS process system service. It can be configured which process needs to be started when an object is inserted, which process needs to be started if an object is deleted, etc.

The *Web crawler* service periodically monitors web sites for modifications. For faster access, a local copy is stored and updated in case of modifications. If a new link is found and this link is qualified for inclusion, the crawler follows this link and inserts the new object. Similar to the Storage service, processes are triggered according to modification events. The only difference is that here it is also necessary to specify at what time and in which intervals the crawler should visit the web sites, inclusion and exclusion patterns need to be defined, etc.

5.2 Search support services

Content-based information retrieval for multimedia data frequently employs features that have been extracted from the objects prior to search to determine the similarity of objects. Depending on the type of features to extract, this can be rather time-consuming. For instance, the analysis of audio files or shot detection in video material can require complex computation, whereas determining the height and width of an image can be performed rather quickly. This class of services encompasses the actual search, but also the services for analysis and characterization of objects, such as for instance feature extractors, that finally allows the application of content-based similarity search.

DelosDLMS offers a generic support for feature-based indexing, allowing different feature extractors to be easily plugged-in. They are implemented as tightly coupled services that share a common interface for extraction and for registering to the system. At registration time, it is also possible to specify the storage format, dependencies on other features, and a preferred distance measure for similarity evaluation. An administration tool helps to automatically generate features extraction processes.

Beside generic support, the DelosDLMS library already includes several feature extraction services for a variety of document and multimedia types, including *textual documents*, *web pages*, *images*, *audio*, *video*, and *3D* objects.

A number of feature extractors were already available in the original ISIS library. The *Hypertext Feature Extractor* analyzes HTML documents for embedded links. It also identifies images and assigns the surrounding text to them. A *Term Frequency Extractor* handles plain text, e.g., the output of the Hypertext Feature Extractor. It returns a list of all found terms and the number of their occurrences. This feature is stored as a set index and used for Boolean and vector space retrieval. The general *Image Feature Extractor component* provides the following features: (1) colour histogram in RGB colour space, (2) colour moments in HCL and Lab colour space, and (3) Gabor texture moments on luminance. All of these can be applied on the image as a whole, 3×3 overlapping rectangles, or "5 fuzzy regions", which basically splits the image into one region starting in each corner and an ellipse in the image centre where a membership function defines for each pixel of the image its influence on the feature of the region. The image can also be extracted from a video file, e.g., a key frame. The *Face Detector* service offers several algorithms to analyze images to identify regions that contain a face. It returns the number of face found and a bounding box for each. The *Audio Feature Extractor* manages features based on beat and pitch.

In addition to these audio services, three different feature sets for audio files specifically suited to describe music have been added. These features sets consider *Rhythm Patterns*, *Statistical Spectrum Descriptors* and *Rhythm Histograms* [25].

The DELOS network has produced several feature sets for the semantic description of *video* resources. One of these sets allows the extraction of semantic descriptors from soccer videos. They represent high-level features like the actions during the game (e.g., pass-forward), the number of players present in a scene or the camera viewpoint [8]. Another feature set, which has been developed within the CoCoMa (Content and Context Aware Multimedia Content Retrieval, Delivery and Presentation) subproject of DELOS, focuses on the representation and management of semantic multimedia annotations and indexing. The DS-MIRF [38] framework allows standard multimedia metadata schemas (MPEG-7) to be enhanced with domain knowledge. It supports the integration of MPEG-7 and OWL-encoded ontologies in order to allow domain specific semantics to be captured. For fine-grained multimedia content delivery, the service extracts in real time video segments that correspond to particular result items identified by the multimedia retrieval service.

3D objects can be indexed and searched for similarity using a feature extraction service based on curvature correlograms as a model for representation [6,7].

After features are extracted from objects, they are indexed to answer queries efficiently. The *indexing service* supports several types of indices. The first two, attribute and set indices, use relational databases. The difference between the two is that attribute indices are built on one particular value, e.g., the size of an object in numbers of bytes. This can be handled easily by the functionality that common relational database management systems (RDBMS) provide. Set indices, in contrast, can contain multiple values, e.g., used to provide text retrieval in vector space model, for which term frequencies and document frequencies are needed to measure relevance. For high-dimensional feature spaces, e.g., if colour histograms of images are used, additional indices are stored outside the RDBMS [27]. The indexing component optimizes and executes queries on these index types. When complex queries are used, it is also necessary to aggregate the results delivered by the individual indices. This task is also performed by this component, since this allows for further performance optimizations [10].

The indexing engine is feature-neutral, which means that new features (and new media types) can be added to DelosDLMS by just plugging in feature-extraction services, without need to change the indexing service.

5.3 Annotation management services

FAST [2] is a system for annotation management. Within FAST, annotations are composite multimedia objects, which can annotate multiple parts 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 regular class digital object which means that it can again be annotated. Beside annotation management functionalities, such as creation, access, and sophisticated search functionalities [3,4], FAST supports collaboration among user by introducing scopes of annotation and groups of users.

5.4 Advanced visualization services

These services provide advanced graphical means to explore and analyze large data collections and query results, as alternative to more traditional views like sorted lists of query result 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. An implementation of the *FastMap* visualization algorithm [19] is available as part of the ISIS library. A *Self Organizing Map* visualization of the feature space [13] helps visualizing a collection in terms of content-based similarity clusters and to locate query objects and results inside the collection. Finally, the *DARE* (Drawing Adequate Representations) System [15,16] adds an interactive 3D view of large high dimensional data sets with structured metadata, and offers visual OLAP capabilities,

allowing the end user to easily aggregate the data in order to get a more abstract view of the data set.

5.5 User query support services

The *relevance feedback* component evaluates the feedback that a user can issue for previously executed queries. This may take place in two different phases:

1. Before the query is executed, it can be reformulated, e.g., by adding more reference objects to the query or by modifying their individual weights.
2. After the query has been executed, it can filter out some unwanted objects or rearrange the position within the result.

SKOSWS [34] is a service for vocabularies represented in SKOS Core Vocabulary [9]. The service works with thesauri or related Knowledge Organization Systems (KOS) represented in the SKOS RDF format. The service allows a search to be augmented by KOS-based vocabulary and semantic resources. Users may browse a concept space to explore and become familiar with specialist terminology or may browse to directly access data linked to concepts. Queries may be expanded by synonyms or by semantically related concepts.

5.6 Non-standard UI support services

These services allow interaction with DL functionalities through non-standard user interfaces. The iPaper infrastructure [29,28] is a *paper-based* interface to DelosDLMS, i.e., it offers core DelosDLMS functionality via a sheet of interactive paper and a corresponding pen. In addition, DelosDLMS features a *natural language* interface which comes from the set of CoCoMa services (see Sect. 5.2). The service is able to analyze natural language sentences and allows digital library users to search for audio-visual content by posing semantic-based criteria through a speech interface. More generally, this service, called OntoNL, is a system for the automatic generation of natural language interfaces for knowledge repositories based only on the domain knowledge provided as an OWL ontology [23].

6 Building DL applications with DelosDLMS

In order to showcase the potential of DelosDLMS and its paradigm for building DL applications, several prototype DL application have been built on top of DelosDLMS starting from the services described in the previous section and a set of large multimedia collections. These applications allow for multimedia, multi-object metadata, and content-based search with relevance feedback over image, audio, video, and 3D objects collections, through standard and non-standard interfaces and with several alternative visualizations of query results.

In this section we describe in more detail how some of these applications have been built, to give a more concrete idea of how the DelosDLMS approach facilitates the production of powerful and efficient DL applications. In order to give an example of the flexibility of the approach, we also show how the same functionality (represented by a process) can be reused for several different applications—possibly in combination with different other functionalities—and that the presentation layer can be changed to adapt it to specific user needs by changing only the front-end, while leaving the underlying processes untouched.

6.1 Content-based search in multimedia collections

A first application which has been built using the DelosDLMS approach provides support for information retrieval in multimedia collections, including content-based retrieval of images, audio and video content, and the combination of any of these media types with sophisticated text retrieval [35].

The screen shots in Fig. 5 show a combined keyword and image similarity search for flowers. Starting point is the query front-end, depicted in Fig. 5a, where keyword and reference image can be specified. Figure 5b then shows the query results for this combined query.

The processes necessary for this type of access to multimedia objects address several different tasks required for creating, maintaining and querying multimedia collections through content-based similarity. To give the flavour of DL process definition in DelosDLMS, we introduce in more detail two of them.

Indexing collections. The sample process illustrated in Fig. 6 shows all necessary activities for inserting a new multimedia object into DelosDLMS. Activities, like Extract Features, may themselves be complex processes, again obtained using a composition of services, e.g., a colour histogram service, a face detection service, and a segmentation service. Activities may be system services, e.g., storing some meta-data tuples in a database, or storing an image in a file system, but also specific DL services like a face detector.

Searching collections. Querying collections is also implemented as a process. This process has already been depicted in Fig. 3 in O'Grape. Each query issued by the user, like the one presented in Fig. 5, is executed as a new process instance. The query process (including user feedback) consists of the steps Query Reformulation (based on relevance feedback the user has issued), Query Execution (index access), and Result Filtering (which may again take user feedback into account).

It is important to note that the process specification just contains the details of all application services it encompasses

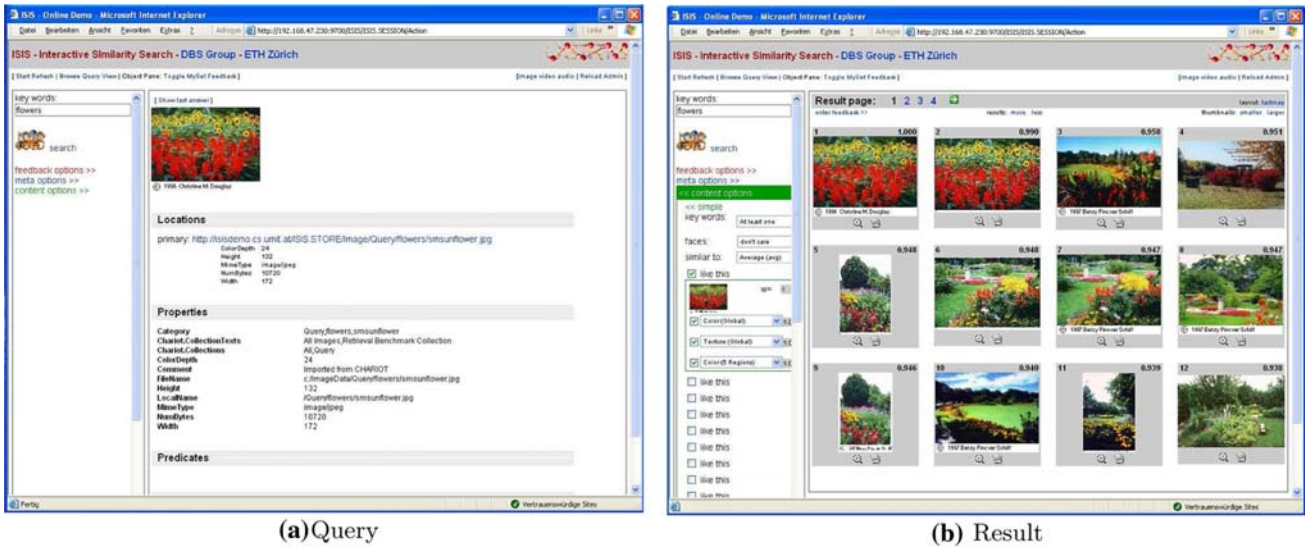

Fig. 5 Combined text and image similarity search

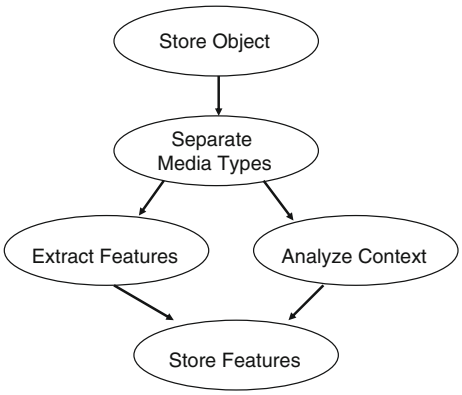

Fig. 6 Sample Process InsertMultimediaObject

(WSDL description for Web services) and the orders within the process. Information on the location of the service providers is not part of the process description. The DelosDLMS services can be easily distributed among several nodes in a network [41], and the actual service providers where the service is invoked are determined at run-time. Hence, each step of the process can be executed by any node providing the required service.

Distribution and parallelism are transparently managed by the underlying OSIRIS middleware. Exploiting them makes DelosDLMS efficient, scalable and reliable, yet easy to design and maintain.

For instance, the heavy computational load which normally arises from content-based queries (which require to evaluate similarity in high-dimensional feature spaces), especially when features of objects need to be extracted on-line, is reduced in DelosDLMS by replicating data on several nodes to serve more requests in parallel (load-balancing) and/or by handling parts of the request on several nodes [11]. Handling replication increases the complexity of updating collections, since the updates of all indexes have to be coordinated to ensure consistency. In DelosDLMS, this is achieved by appropriate system processes, which run automatically to guarantee consistency over several replicas of the index.

Similarly, the extraction of features itself can be a time-consuming task. For this reason, monitoring constantly changing collections and providing access can be challenging as well. If the insertion of multimedia objects can be divided in several sub-tasks and those can be executed on different nodes while using an infrastructure ensuring correctness of the distributed execution, this can improve the performance significantly [39]. The process used in DelosDLMS to insert new objects (depicted in Fig. 6) follows this idea of distributing effort across different nodes in the system.

DelosDLMS is efficiently searching and maintaining a collection of more than 600,000 images used within ISIS in the ETHWorld project, the virtual campus of ETH Zurich. The images have been extracted from web sites of the university and all its institutes.

6.2 Flexible DL application reengineering with DelosDLMS

A key feature of the DelosDLMS approach to DL application building is its modularity, that allows for flexibility and reuse. As an example of this, we describe here in short how some of the functionalities described in the previous section were integrated into completely different front-ends, including non-conventional ones. Some of these user interfaces are specialized input devices, while others are software components developed independently as stand-alone DL applications.

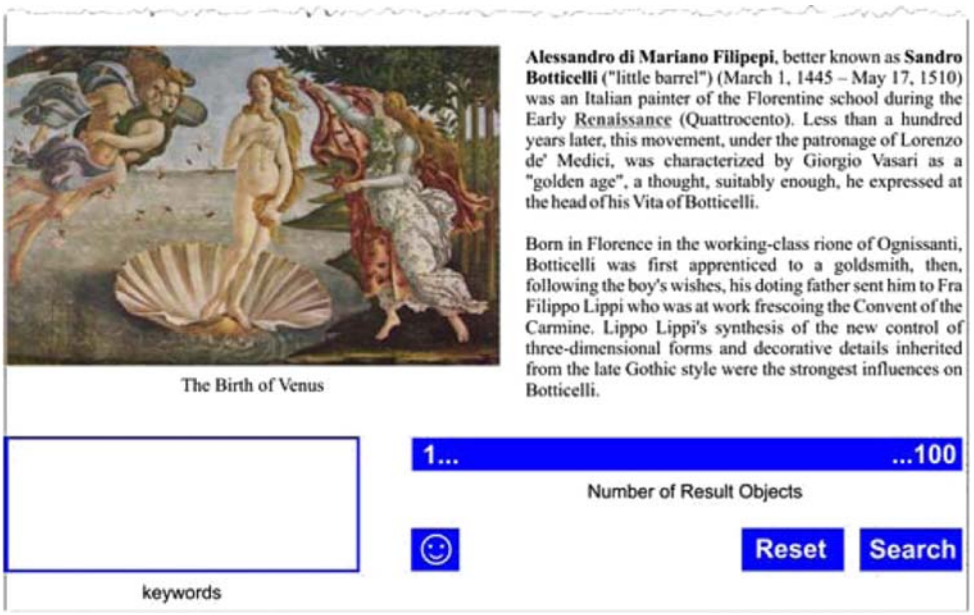

Fig. 7 Excerpt of the Interactive Museum Guide

Interactive Museum Guide. The paper based interface briefly introduced in the first part of this section is a system 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.

Starting from this infrastructure, an interactive printed museum guide was developed as a physical user interface to the DelosDLMS. In the case of the interactive museum guide shown in Fig. 7, printed text, images, blue "paper buttons" and other interface components have been associated with specific query functionality. the interface allows for keyword and content-based retrieval. Users can either select keywords by pointing the pen at spcific underlined words within the text, or they can enter them in a special keyword input area (any keywords entered in the keyword area are transformed to a text string by an intelligent character recognition process). An arbitrary number of keywords may be aggregated by selecting parts of the text or writing keywords. Queries based on image similarity (alone, or in combination with keyword based searches) are possible by touching one or more images in the museum guide with the digital pen. The paper buttons allow to start the search, configure specific query parameters such as the preferred size of the result set or a face recognition functionality, and resetting the parameters of the query.

The application supporting the interactive museum guide is built on DelosDLMS. The sophisticated query functionalities available in ISIS are, as already described, available as specialized processes. In order to enable them for the paper based interface, the only required action was to write a component that records user's interaction with the museum guide and invokes the appropriate processes with the right parameters.

Daffodil. The digital library system Daffodil [24], developed by the University of Duisburg-Essen, is a front-end system for DLs, 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 DLs. 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. A natural improvement to the funcitonality already offered by Daffodil was to make available to its user the content-based query capabilities of 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; and (2) query by keyword: given a specific keyword, the system is able to search for pictures indexed with this keyword. Both functions allow query refinement by user feedback. The two functions were made available by implementing a specialized tool for the Daffodil system.

As in the previous case, the tool is a thin front-end to the same query processes defined for ISIS and supported by the OSISIS environment.

MedioVis. MedioVis [20,26] is a visual information seeking system developed at the University of Konstanz. It demonstrates novel ways of information seeking in DLs for novice and non-expert users by integrating novel interaction and visualization techniques in a consistent user interface design. A central design goal of the MedioVis interface is to offer interaction design that supports realistic human search behaviour. Therefore not only analytical queries but also browsing-oriented and opportunistic retrieval strategies are supported, for example by allowing visual overviews, visual filtering, zooming into details, and "surfing" between the search results' metadata and full text. Visually handling metadata and content is possible by using novel zoomable user interface components and visualizations (e.g. HyperGrid [22] or HyperScatter) which allow a direct manipulation of the displayed level of detail and therefore gradually blur the boundaries between metadata, full text and external data sources. Offering the user new search options, e.g. to find paintings or portraits similar to the ones the user has selected, furthermore supported natural information seeking strategies. As in the case of Daffodil, integrating content-based image retrieval functionality into Mediovis with the DelosDLMS approach allowed to dramatically enhance its functionality in an easy way, and to blend the functionality already offered by ISIS with the rich, user centric experience provided by Mediovis.

7 Conclusion

Digital Libraries have undergone significant changes over the last years. They are more and more becoming the universal knowledge repositories and communication conduits. People access, analyze, evaluate, enhance, and exchange all forms of information in their daily and professional life.

A particular challenge is the ever increasing number of data sources in a DL, reaching from traditional databases and large (multimedia) document collections, information sources contained in web pages, down to information systems in mobile devices as they will occur in a pervasive computing environment. Therefore not only the immense amount of information demands new thoughts but also the number of different information sources. Essentially, their coordination poses a great challenge for the development of future tools that will be suitable to access, process, and maintain information.

Recent trends show that DL functionality is more and more available in a modular way, rather than by means of monolithic systems. Thus, the realization of the DELOS vision for future DLs requires a sophisticated DL infrastructure to support the combination of existing DL functionality so as to best meet the needs of the different users and user communities. Such an infrastructure must provide convenient tools for accessing information, for developing applications for analyzing, mining, classifying, and processing information, and for transactional processes that ensure consistent propagation of information changes and simultaneous invocations of several services within a transactional workflow.

In this paper, we have presented DelosDLMS, a joint effort of a large number of partners within the DELOS Network of Excellence. Driven by the DELOS vision for DLs, DelosDLMS is the implementation of a prototype of a next-generation DLMS that jointly addresses all the challenges presented above. It supports the joint management of different media types by combining text and audio-visual searching. It offers personalized browsing using new information visualization and relevance feedback tools and it provides novel user interfaces by which retrieved information is presented to the user. Support for cooperation between users is added, for instance, by services for annotating objects and for providing shared access to this user-defined content. Most importantly, the system is open, extensible, and supports the integration of distributed content and services into a single process-based application. It currently features a rich set of DL services, but can be further enriched by plugging in additional specialized DL functionality as services.

Acknowledgements The success of DelosDLMS has only been possible due to the highly valuable contributions of many partners in the DELOS network and the excellent collaboration. Special thanks go to: Maristella Agosti (University of Padua), Stefano Berretti (University of Florence), Marco Bertini (University of Florence), Gert Brettlecker (University of Basel), Alberto del Bimbo (University of Florence), Tiziana Catarci (University of Rome "La Sapienza"), Stavros Christodoulakis (Technical University of Crete), Tom Crecelius (Max-Planck Institut Saarbrücken), Nicola Ferro (University of Padua), Norbert Fuhr (University of Duisburg-Essen), Nektarios Gioldasis (Technical University of Crete), Hans-Christian Jetter (University of Konstanz), Daniel Keim (University of Konstanz), Claus-Peter Klas (University of Duisburg-Essen), Thomas Lidy (Technical University of Vienna), Sebastian Michel (Max-Planck Institut Saarbrücken), Moira Norrie (ETH Zurich), Paola Ranaldi (University of Basel), Andreas Rauber (Technical University of Vienna), Harald Reiterer (University of Konstanz), Giuseppe Santucci (University of Rome "La Sapienza"), Marc Scholl (University of Konstanz), Tobias Schreck (University of Konstanz), Beat Signer (ETH Zurich), Michael Springmann (University of Basel), Doug Tudhope (University of Glamorgan), Gerhard Weikum (Max-Planck Institut Saarbrücken).

References

1. ACM digital library. http://portal.acm.org/dl.cfm
2. Agosti, M., Ferro, N.: A System architecture as a support to a flexible annotation service. In: Proceedings of the 6th Thematic DELOS Workshop, pp. 147–166 (2005a)
3. Agosti, M., Ferro, N.: Annotations as context for searching documents. In: Proceedings of the CoLIS'2005. Glasgow (2005b)

4. Agosti, M., Ferro, N.: Search strategies for finding annotations and annotated documents: the FAST service. In: Proceedings of the FQAS'2006. Milan (2006)
5. Agosti, M., Berretti, S., Brettlecker, G., del Bimbo, A., Ferro, N., Fuhr, N., Keim, D.A., Klas, C.P., Lidy, T., Milano, D., Norrie, M.C., Ranaldi, P., Rauber, A., Schek, H.J., Schreck, T., Schuldt, H., Signer, B., Springmann, M.: DelosDLMS—the integrated DELOS digital library management system. In: Proceedings of the 1st DELOS Conference, pp. 36–45. Tirrenia, Italy (2007)
6. Antini, G., Berretti, S., del Bimbo, A., Pala, P.: Curvature correlograms for content based retrieval of 3D objects. In: Proceedings of the ICIAP'2005. Cagliari (2005a)
7. Antini, G., Berretti, S., del Bimbo, A., Pala, P.: Retrieval of 3D objects using curvature correlograms. In: Proceedings of the ICME'05. Amsterdam (2005b)
8. Ballan, L., Bertini, M., del Bimbo, A., Nunziati, W.: Soccer players identification based on visual local features. In: Proceedings of the 6th ACM International Conference on Image and Video Retrieval (CIVR 2007), pp. 258–265. ACM, Amsterdam (2007)
9. Binding, C., Tudhope, D.: KOS at your service: programmatic access to knowledge organisation systems. J. Digit. Inf. **4**(4) (2004)
10. Böhm, K., Mlivoncic, M., Schek, H.J., Weber, R.: Fast evaluation techniques for complex similarity queries. In: Proceedings of the VLDB'01. San Francisco (2001a)
11. Böhm, K., Mlivoncic, M., Weber, R.: Quality-aware and load-sensitive planning of image similarity queries. In: Proceedings of the ICDE'2001. Washington (2001b)
12. Brettlecker, G., Milano, D., Ranaldi, P., Schek, H.J., Schuldt, H., Springmann, M.: ISIS & OSIRIS: a process-based digital library application on top of a distributed process support middleware. In: Proceedings of the 1st DELOS Conference. Springer LNCS, Tirrenia (2007)
13. Bustos, B., Keim, D., Panse, C., Schreck, T.: 2D maps for visual analysis and retrieval in large multi-feature 3D model databases. In: Proceedings of the IEEE Visualization Conference (VIS'2004). Austin (2004)
14. Candela, L., Castelli, D., Pagano, P., Thanos, C., Ioannidis, Y., Koutrika, G., Ross, S., Schek, H.J., Schuldt, H.: Setting the foundations of digital libraries: the DELOS Manifesto. D. Libr. Mag. **13**(3/4) (2007)
15. Catarci, T., Santucci, G.: The prototype of the dare system. In: Proceedings of the SIGMOD Conference, p. 609 (SIGMOD 2001), Santa Barbara (2001)
16. Catarci, T., Santucci, G., Costabile, M.F., Cruz, I.F.: Foundations of the dare system for drawing adequate representations. In: DANTE, pp. 461–470 (1999)
17. Children's Digital Library. http://www.storyplace.org
18. Digital libraries: future directions for a European research program. DELOS brain-storming report, San Cassiano, June 2001
19. Faloutsos, C., Lin, K.I.: Fastmap: a fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. In: SIGMOD Conference, pp. 163–174 (1995)
20. Grün, C., Gerken, J., Jetter, H., König, W., Reiterer, H.: MedioVis—a user-centred library metadata browser. In: Proceedings of the 9th European Conference on Research and Advanced Technology for Digital Libraries (ECDL'2006), pp. 174–185. Springer LNCS, Vienna (2006)
21. Ioannidis, Y.: Digital libraries at a crossroads. Int. J. Digit. Libr. **5**(4), 255–265 (2005)
22. Jetter, H., Gerken, J., König, W., Grün, C., Reiterer, H.: HyperGrid—accessing complex information spaces. In: People and Computers XIX-The Bigger Picture, Proceedings of HCI 2005. Springer, Edinburgh (2005)
23. Karanastasi, A., Zotos, A., Christodoulakis, S.: User interactions with multimedia repositories using natural language interfaces—OntoNL: an architectural framework and its implementation. J. Digit. Inf. Manage. (JDIM) **4**(4) (2006)
24. Klas, C.P., Fuhr, N., Schaefer, A.: Evaluating strategic support for information access in the DAFFODIL system. In: Proceedings of the 8th European Conference on Research and Advanced Technology for Digital Libraries (ECDL'2004). Bath (2004)
25. Lidy, T., Rauber, A.: Classification and clustering of music for novel music access applications. In: Machine Learning Techniques for Multimedia. Springer, Heidelberg (2008)
26. Mediovis project. http://hci.uni-konstanz.de/MedioVis
27. Mlivoncic, M., Weber, R.: A filter and refinement approach to RBIR. In: Proceedings of the 5th DELOS Thematic Workshop. Chania (2003)
28. Norrie, M.C., Palinginis, A., Signer, B.: Content publishing framework for interactive paper documents. In: ACM Symposium on Document Engineering (2005)
29. Norrie, M., Signer, B., Weibel, N.: General framework for the rapid development of interactive paper applications. In: Proceedings of CoPADD'2006. Banff (2006)
30. Schek, H.J., Schuldt, H.: DelosDLMS—infrastructure for the next generation of digital library management systems. ERCIM, Spec Issue Eur. Digit. Libr. 66 (2006)
31. Schuldt, H., Alonso, G., Beeri, C., Schek, H.J.: Atomicity and isolation for transactional processes. ACM Trans. Database Syst. **27**(1), 63–116 (2002)
32. Schuler, C., Türker, C., Schek, H.J., Weber, R., Schuldt, H.: Scalable peer-to-peer process management. Int. J. Bus. Process Integr. Manage. (IJBPIM) **1**(2), 129–142 (2006)
33. Schuler, C., Schuldt, H., Türker, C., Weber, R., Schek, H.J.: Peer-to-peer execution of (transactional) processes. Int. J. Coop. Inf. Syst. **14**(4), 377–406 (2005)
34. Simple Knowledge Organisation Systems. W3C Semantic Web Deployment Working Group. http://www.w3.org/2004/02/skos/
35. Springmann, M.: A novel approach for compound document matching. Bull. IEEE Tech. Comm. Digit. Libr. (TCDL) **2**(2) (2006)
36. The Perseus Digital Library. http://www.perseus.tufts.edu
37. The Online Books Page. http://www.digital.library.upenn.edu/books
38. Tsinaraki, C., Polydoros, P., Christodoulakis, S.: Interoperability support between mpeg-7/21 and owl in ds-mirf. IEEE Trans. Knowl. Data Eng. **19**(2), 219–232 (2007)
39. Weber, R., Schek, H.J.: A distributed image-database architecture for efficient insertion and retrieval. In: Proceedings of MIS'99, pp. 48–55. Indian Wells (1999)
40. Weber, R., Schek, H.J., Blott, S.: A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In: Proceedings VLDB'98, pp. 194–205. New York City (1998)
41. Weber, R., Bolliger, J., Gross, T.R., Schek, H.J.: Architecture of a networked image search and retrieval system. In: Proceedings of CIKM'99. Kansas City (1999)