Supporting Information Access in Next Generation Digital Library Architectures[*]

Predrag Knežević[1], Bhaskar Mehta[1], Claudia Niederée[1], Thomas Risse[1], Ulrich Thiel[1], and Ingo Frommholz[2]

[1] Fraunhofer IPSI
Integrated Publication and Information Systems Institute
Dolivostrasse 15, 64293 Darmstadt, Germany
{knezevic|mehta|niederee|risse|thiel}@ipsi.fhg.de

[2] University of Duisburg-Essen
Institute of Informatics and Interactive Systems
D-47048 Duisburg, Germany
ingo.frommholz@uni-due.de

Abstract. Current developments on Service-oriented Architectures, Peer-to-Peer and Grid computing promise more open and flexible architectures for digital libraries. They will open DL technology to a wider clientele, allow faster adaptability and enable the usage of federative models on content and service provision. These technologies rise new challenges for the realization of DL functionalities, which are rooted in the increased heterogeneity of content, services and metadata, in the higher degree of distribution and dynamics, as well as in the omission of a central control instance. This paper discusses these opportunities and challenges for three central types of DL functionality revolving around information access: metadata management, retrieval functionality, and personalization services.

1 Introduction

Currently, there is a considerable amount of R&D activity in developing viable strategies to use innovative technologies and paradigms like Peer-to-Peer Networking, Grid, and Service-oriented Architectures in digital libraries (see e.g. the European Integrated Projects BRICKS [1] and DILIGENT [2]). The promise is that these efforts will lead to more open and flexible digital library architectures that:

- open up digital library (DL) technology to a wider clientele by enabling more cost-effective and better tailored digital libraries,
- allow faster adaptability to developments in DL services and IT technologies, and
- enable usage of dynamic federative models of content and service provision involving a wide range of distributed content and service providers.

The use of Service-oriented Architectures, Grid infrastructures, and the Peer-to-Peer approach for content and service provision has implications for the realization of

[*] This work is partly funded by the European Commission under BRICKS (IST 507457), COLLATE (IST-1999-20882), DILIGENT (IST 004260) and VIKEF (IST 507173)

enhanced DL functionality. These implications are mainly rooted in increased heterogeneity of content, services and metadata, in the higher degree of distribution and dynamics, as well as in the omission of a central control instance. On one hand, these are opportunities for better and more multifarious DL services; on the other hand, these are new challenges to ensuring long-term, reliable, and quality-ensured DL service provision that also exploits the technology promises. This paper discusses these opportunities and challenges for three central types of DL functionality revolving around information access: metadata management, retrieval functionality, and personalization services.

The rest of this paper is structured as follows: Section 2 presents the key ideas of next generation DL architectures based on exemplary RTD projects. Section 3 discusses how these new ideas influence information access in the areas of metadata management, information retrieval, and personalization support. Related work in these areas is considered in section 4. The paper concludes with a summary of the paper's key issues.

2 Next Generation Digital Library Architectures

Current plans for next generation DL architectures are aiming for a transition from the DL as an integrated, centrally controlled system to a dynamic configurable federation of DL services and information collections. This transition is inspired by new technology trends and developments. This includes technologies like Web services and the Grid as well as the success of new paradigms like Peer-to-Peer Networking and Service-oriented Architectures. The transition is also driven by the needs of the ”DL market”:

- better and adaptive tailoring of the content and service offer of a DL to the needs of the respective community as well as to the current service and content offer;
- more systematic exploitation of existing resources like information collections, metadata collections, services, and computational resources;
- opening up of DL technology to a wider clientele by enabling more cost-effective digital libraries.

To make these ideas more tangible we discuss three RTD projects in the field and discuss the relationship to upcoming e-Science activties.

2.1 Virtual Digital Libraries in a Grid-based DL Infrastructure

DILIGENT[3] is an Integrated Project within the IST 6th Framework Programme. It's objective is ”to create an advanced test-bed that will allow members of dynamic virtual e-Science organizations to access shared knowledge and to collaborate in a secure, coordinated, dynamic and cost-effective way.”

The DILIGENT testbed will enable the dynamic creation and management of Virtual Digital Libraries (VDLs) on top of a shared Grid-enabled DL infrastructure, the DILIGENT infrastructure. VDLs are DLs tailored to the support of specific e-Science communities and work groups. For creating a VDL, DL services, content collections, metadata collections are considered as Grid resources and are selected, configured, and

[3] DILIGENT - A DIgital Library Infrastructure on Grid ENabled Technology

integrated into processes using the services of the DILIGENT infrastructure. This infrastructure builds upon an advanced underlying Grid infrastructure as it is currently evolving e.g. in the EGEE project[4].

Such a Grid infrastructure will already provide parts of the functionality required for DILIGENT. This includes the dynamic allocation of resources, support for cross-organizational resource sharing, and a basic security infrastructure. For effectively supporting DLs, additional services like support for redundant storage and automatic data distribution, metadata broker, metadata and content management, advanced resource brokers, approaches for ensuring content security in distributed environments and the management of content and community workflows are rquired in addition to services that support the creation and management of VDLs. A further project challenge are systematic method to make the treasure of existing DL services and collections utilizable as Grid resources in the DILIGENT infrastructure.

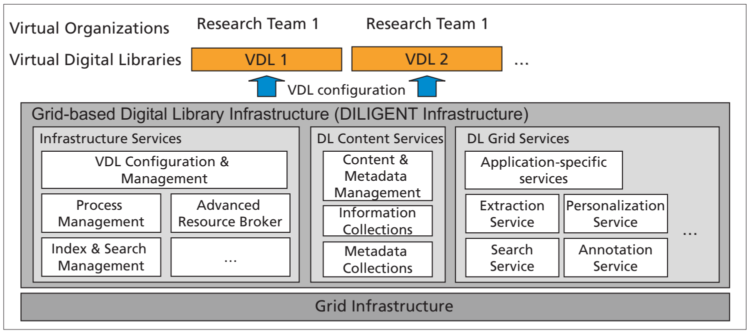

Fig. 1. DILIGENT Virtual Digital Library Infrastructure

Figure 1 shows an architecture overview of the DILIGENT infrastructure. Based on the Grid infrastructure it contains three types of services. The *Infrastructure Services* are a group of services that are specific for operating the infrastructure. This group contains the services for the configuration and management of VDLs. The *DL Grid Services* are a set of specific library services. On one hand, existing DL services are wrapped as Grid Services or adapted to the new Grid-based environments. On the other hand, new services are implemented that specifically take into account the operation environment, e.g. services for distributed search. Besides generic services like annotation and search services, this group also contains application specific services. Finally, the *DL Content Services* include services for content and metadata management as well as existing information and metadata collections wrapped as Grid services. This DILIGENT infrastructure is used to create VDLs in support of virtual organizations.

[4] http://public.eu-egee.org

The DILIGENT project will result in a Grid-enabled DL testbed that will be validated by two complementary real-life application scenarios: one from the Cultural Heritage domain and one from the environmental e-Science domain.

2.2 Service-oriented and Decentralized DL Infrastructure

The aim of the BRICKS[5] project [1] is to design, develop and maintain an open user and service-oriented infrastructure to share knowledge and resources in the Cultural Heritage domain. The target audience is very broad and heterogeneous and involves cultural heritage and educational institutions, research community, industry, and citizens. Typical usage scenarios are integrated queries among several knowledge resource, e.g. to discover all Italian artefacts from renaissance in the European museums. Another example is to follow the life cycle of historic documents, whose manual copies are distributed all over Europe. These examples are specific application, which are running on top of the BRICKS infrastructure.

The BRICKS infrastructure uses the Internet as a backbone and has to fulfil the following requirements:

- Expandability, which means the ability to acquire new services, new content, or new users, without any interruption of service.
- Scalability, which means the ability to maintain excellence in service quality, as the volumes of requests, of content and of users increase.
- Availability, which means the ability to operate in a reliable way over the longest possible time interval.
- Graduality of Engagement, which means the ability to offer a wide spectrum of solutions to the content and service providers that want to become members of BRICKS.
- Interoperability, which means the ability to make available services to and exploit services from other digital libraries.

In addition, the user community has the economic requirement to be low-cost. This means (1) that an institution should be able to become a BRICKS member with minimal investments, and (2) that the maintenance costs of the infrastructure, and in consequence the running costs of each BRICKS member, are minimized.

Interested institution should not invest much additional money in its already existing infrastructure to become a member of BRICKS. In the ideal case the institution should only get the BRICKS software distribution, which will be available for free, install it, connect to the internet and become a BRICKS member. This will already gives the possibility to search for content and access some services. For sure, additional work is necessary to integrate and adapt existing content and services to provide them in BRICKS.

Also, the BRICKS membership will be flexible, such that parties can join or leave the system at any point in time without administrative overheads. To minimize the maintenance cost of the infrastructure any central organization, which maintains e.g. the service directory, should be avoided.

[5] BRICKS - Building Resources for Integrated Cultural Knowledge Services

With respect to access functionality, BRICKS provides appropriate task-based functionality for indexing/annotation and collaborative activities e.g. for preparing a joint multimedia publication. An automatic annotation service will enable users to request background information, even if items have not been annotated by other users yet. By selecting appropriate items, such as definitions of concepts, survey articles or maps of relevant geographical areas, the service exploits the currently focussed items and the user's goals expressed in the user profile. In addition, the linking information, which is generated dynamically, must be integrated into the documents. The design of the access functionality is influenced by our experiences in the 5th Framework project COLLATE.

2.3 COLLATE: A Web-based environment for document-centered collaboration

Designed as a content- and context-based knowledge working environment for distributed user groups, the COLLATE system supports both individual work and collaboration of domain experts with material in the data repository. The example application focuses on historic film documentation, but the developed tools are designed to be generic and as such adaptable to other content domains and application types. This is achieved by model-based modules.

The system supports collaborative activities such as creating a joint publication or assembling and creating material for a (virtual) exhibition, contributing unpublished parts of work in the form of extended annotations and commentaries. Automatic indexing of textual and pictorial parts of a document can be invoked. Automatic layout analysis for scanned documents can be used to link an annotation of individual segments. As a multifunctional means of in-depth analysis, annotations can be made individually but also collaboratively, for example in the form of annotation of annotations, collaborative evaluation, and comparison of documents. Through interrelated annotations users can enter into a discourse on the interpretation of documents and document passages.

The COLLATE collaboratory is a multifunctional software package integrating a large variety of functionalities that are provided by cooperating software modules residing on different servers. It can be regarded as a prototypical implementation of a decentralized, Service-oriented DL architecture which serves as a testbed for the collaborative use of documents and collections in the Humanities. The collaborative creation of annotation contexts for documents offers new opportunities for improving the access functionality, as we will illustrate later on.

2.4 Next Generation DL Architectures and e-Science

Scientific practice is increasingly reliant on data-intensive research and international collaboration enabled by computer networks. The technology deployed in such scenarios allows for high bandwidth communication networks, and by linking computers in ”Grids” places considerably more powerful computing resources is at their disposal than a single institution could afford. If we view e-Science as being primarily motivated up to now by notions of resource sharing for computationally intensive processes (e.g. simulations, visualisation, data mining) a need is emerging for new approaches, brought up by ever more complex procedures, which, on the one hand, assume the reuse of workflows, data and information and, on the other hand, should be able to support collaboration in virtual teams. Future concepts of e-Science will be less focussed on data and computing resources, but will include services on the knowledge and organizational levels as well. Embedding future DL architectures in an emerging e-Science infrastructure will meet these requirements by providing access to information and knowledge sources, and appropriate collaboration support on top of the Grid-based infrastructure.

3 Information Access in Next Generation DL Architectures

A decentralized, service-oriented architecture poses new challenges to the technologies employed for information access. DLs based on such an architecture should, for example, not only provide access and retrieval functionality for the documents residing on the local peer, but should also consider other peers which might host relevant document w.r.t. a query. In the following, we will outline possible approaches for enhanced services for information access. Such services will utilize the functions of a decentralized metadata management ensuring the availability of all documents (and their parts) while reducing overhead costs. Retrieval functions can be improved by taking into account the annotational contexts of documents emerging for the collaborative process of interpreting and discussing items of interests by a group of users. In addition, individual users‘ contexts can be used to personalize the access services.

3.1 Decentralized Metadata Management

DLs usually like to keep content under control in their local repositories. On the contrary, metadata should be available for all parties, stored in some central place accessible for everybody. Decentralized architectures by definitions avoid having central points, for the following reasons: they are candidate single point of failure and performance bottleneck. Therefore, metadata must be spread in the community. A naïve approach for metadata searching would be to distribute queries to all members, but it is obvious that the solution is unscalable. Hence, efficient metadata access and querying are very important challenges within the new decentralized settings.

Our proposal to these challenges is a decentralized Peer-to-Peer datastore that will be used for managing XML-encoded metadata. It balances resource usage within the community, has high data availability (i.e. data are accessible even if creator disappears from the system, e.g. system fault, network partitioning, or going offline), is updateable (i.e. stored data can be modified during the system lifetime), and supports a powerful query language (e.g XPath/XQuery).

XML documents are split into finer pieces that are spread within the community. The documents are created and modified by the community members, and can be accessed from any peer in a uniform way, e.g. a peer does not have to know anything about the data allocation. Uniform access and balanced storage usage are achieved by using a DHT (Distributed Hash Table) Overlay [3] and having unique IDs for different document parts.

Figure 2 shows the proposed database architecture. All layers exist on every peer in the system. The datastore is accessed through the P2P-DOM component or by using

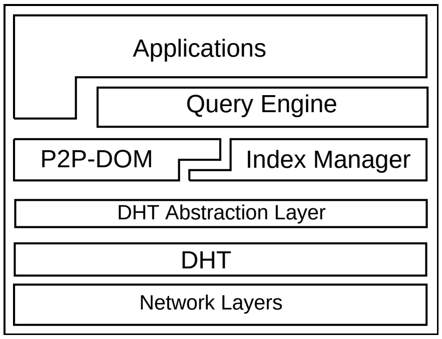

Fig. 2. Decentralized XML Storage Architecture

the query engine. The query engine could be supported by an optional index manager that maintains indices.

P2P-DOM is the core system layer. It exports a large portion of the DOM interface to the upper layers, and maintains a part of a XML tree in a local store (e.g. files, database). P2P-DOM serves local requests (adding, updating and removing of DOM-tree nodes) and requests coming from other peers through a Distributed Hash Table (DHT) [4–6] overlay, and tries to keep the decentralized database in a consistent state. In order to make the DHT layer pluggable, it is wrapped in a tiny layer that unifies APIs of particular implementations, so the upper layer does not need to be modified. A more detailed discussion about the proposed approach, challenges and open issues can be found in [7].

In the rest of the subsection, we are giving more details how the proposed datastore could be used for managing service metadata, which are an additional type of DL metadata introduced by Service-oriented Architectures.

Service metadata describe service functionalities, interfaces and other properties. These meta-information are usually encoded by using WSDL (Web Service Description Language [8]) and published to an UDDI (Universal Description, Discovery and Integration [9]) service directory. Service discovery queries are usually more complex than simple name matching, i.e. they contain qualified, range and/or boolean predicates.

In order to realize a decentralized service directory with advanced query mechanisms, the community of service providers will create and maintain in the decentralized P2P data store a pool of the service descriptions. Every service will be able to modify its description during the lifetime and to search for needed services. Query execution will be spread at many peers, the query originator will only get the final result back.

At the same time, due to uniform data access, new community members can start using the service directory immediately after joining the system without additional setup and administration. A member decision to leave the community will not make any influence for the rest of the system, because data are replicated. Even if network partitioning happens, the service directory would provide access to service metadata available in the partition allowing some parties to continue with work without interruption.

For details about the use of the decentralized datastore in other scenarios see [10].

3.2 Decentralized Context-based Information Retrieval

DLs based on a decentralised architecture should not only provide access and retrieval functionality for the documents residing on the local peer, but should also consider other peers which might host relevant document w.r.t. a query. It is clear that for a scenario like described above appropriate search functionality has to be defined. In the following, we will outline possible approaches for enhanced retrieval services.

Services In order to be able to abstract from the underlying infrastructure, retrieval functionality should be implemented as a service with a predefined API and behaviour. This has the advantage that other peers are able to query the local repository, which is an important feature for enabling P2PIR. An example Web Service specification for search and retrieval is SRW[6]. It considers content-based retrieval functionality, but lacks context-based features as proposed above. When performing retrieval based on the annotation context (see below), such context information should be contained in the result set in order to elucidate why an item was retrieved. So a common API for queries, results and indexing requests has to be identified which is capable of taking advanced queries and context information into account.

Annotation Context

According to the considerations in [11], annotations can be discussed from different viewpoints. From a syntactic point of view, annotations are metadata, since they are connected to the objects they annotate. From their semantics, annotations can either contain content about content (like we find it, for instance, in reviews, recensions, and judgements about documents) or additional content manifesting itself in, e.g., elaborations or augmentations of the content at hand, but also in links that connect documents and annotations with other objects. Interpretations, like we find them in the cultural domain, are an example of annotations conveying both content about content and additional content. Regardless of their semantics, annotations are dialogue acts following certain pragmatics. These pragmatics describe the intention behind a user's statement. This means that annotations consist of certain commmunicative acts [12], which can be, e.g., assertives, directives, and commissives[7]. In both digital libraries and collaboratories, annotations can play a beneficial role w.r.t. certain aspects of such systems. They support authoring and editing, access and retrieval, effective use, interaction, and sharing of data and resources.

Annotations can be either manually or automatically created. Manual annotations range from personal to shared to public ones. They can include personal notes, e.g., for comprehension, and whole discussions about documents [13, 14]. Annotations are building blocks for collaboration. In a distributed, decentralized environment, especially shared and public annotations pose a challenge to the underlying services. Users can create shared and public annotations residing on their peers, but this data has to be spread to other peers as well.

By automatic annotations, we mean the automatic creation and maintenance of annotations consisting of links to and summaries of documents on other peers which are similar to documents residing on the local peer. Such annotations constitute a context

[6] `http://www.loc.gov/z3950/agency/zing/srw/`

[7] Even creating a link between objects is a communicative act, since one makes an assertion about the relationship of these objects, or at least that there is such a relationship at all.

in which documents on a peer are embedded. Dynamic links raise the degree to which the users' work is efficiently supported by the digital library [15]. They provide the opportunity to create comprehensive answers to submitted queries, an idea which is also stated in [16]. For each document, agents could be triggered to periodically update the information at hand, similar to the internal linking methods like similarity search, enrichment and query generation proposed in [15]. P2PIR methods can possibly be applied for this. The underlying assumption is that a user stores potential interesting documents on her peer and is interested in similar publications. Automatic annotations can be created w.r.t. several aspects. For instance, topical similar documents can be sought after. Another interesting kind of automatic annotation can be extracted from the surroundings of a citation. If a document residing on another peer cites a document on the local peer, the surroundings of this citation usually contain some comments about the cited document (similar as reported in [17]). Since only annotations to documents residing on the peer are created, storage costs can be kept low. Regular updates performed by agents keep the user informed.

Annotations, either manual or automatic ones, constitute a certain kind of *document context*. Annotation-based retrieval methods [13] can employ the annotation context without the need to actually access other peers. Since annotations, being manually or automatically created, contain additional information about the document, we assert that annotation-based retrieval functions boost retrieval effectiveness. Future work will show if this assumption holds. Using annotations for information retrieval in a decentralized environment has the advantage that annotations are locally available, but reflect information lying on other peers. In this way, annotations create new access structures which help adressing problems arising when performing information retrieval on an underlying P2P infrastructure.

3.3 Cross-Service Personalization

Personalization approaches in DLs dynamically adapt the community-oriented service and content offerings of a DL to the preferences and requirements of individuals [18]. They enable more targeted information access by collecting information about users and by using these user models (also called user profiles) in information mediation.

Personalization typically comes as an integral part of a larger system. User profiles are collected based on a good knowledge about the meaning of user behavior and personalization activities are tailored to the functionality of the respective system. Within a next-generation distributed DL environment, which is rather a dynamic federation of library services than a uniform system, there are at least two ways to introduce personalization. In the simple case, each service component separately takes care of its personalization independently collecting information about users. A more fruitful approach, however, is to achieve personalization across the boundaries of individual services, i.e., cross-system or, more precisely, cross-service personalization. In this case, personalization relies on a more comprehensive picture of the user collected from his interaction with different library services.

Cross-service Personalization Challenges Cross-service personalization raises the following challenges:

- How to bring together the information collected about a user and his interactions with the different services in a consistent way?
- How to make make up-to-date, aggregated user information about the user available to the different services, i.e. how to manage, update, and disseminate user models to make them accessible to the different services?
- How to support (at least partial) interpretation of the user model in a heterogeneous, and dynamically changing DL service environment?

This requires a shared underlying understanding of the user model. Furthermore, it raises issues of privacy and security, since personal data is moved around in a distributed system. It has to be taken into account that the privacy concerns of the user might be increased by the fact that the information collected from the interaction with different services is combined. This adds an additional challenge for cross-system and cross-service personalization, namely to give the user some control over the information that is collected about him.

Approaches to Cross-Service Personalization We identified two principle approaches which differ from each other in their architecture:

Adaptor approach: The adaptor approach relies on the ideas of wrapper architectures. A kind of wrapper is used to translate information access operations into personalized operations based on the information collected in the context passport.
The advantage of this approach is that personalization can also be applied to services that themselves do not support personalization. The disadvantage is that every service will need its own wrapper. Unless there is a high degree of standardization in service interfaces, creating wrappers for every individual services may not be practical and does not scale well in dynamic service environments.

Connector approach: In contrast to the adaptor approach, the connector approach relies on the personalization capabilities of the individual services. It enables the bidirectional exchange of data collected about the user between the context passport and the personalization
component of the respective service. The context passport is synchronized with individual user models/profiles maintained by services. The advantage here is that personalization of one service can benefit from the personalization efforts of another.

A flexible and extensible user model that can capture various characteristics of the user and his/her context is in the core of both approaches. An example of such a model, that is rather a metamodel for describing different user models is the UUCM model described in [19].

As an operationalization of such a model we developed the idea of a *context passport*. A context passsport accompanies the user and is ”presented” to services to enable personalized support. The context passport [19] is positioned as a temporal memory for information about the user. It covers an extensible set of facets modeling different user model dimensions, including cognitive pattern, task, relationship, and environment dimension. The selection of the dimensions is based on user models in existing personalization approaches and on context modeling approaches. The context passport acts as an aggregated service-independent user profile with services receiving personalization data from the context passport. Services also transfer information to the context passport based on relevant user interaction which add up-to-date information to the user's context. Here it is important that the information is exchanged on an aggregation level that is meaningful outside the individual service. The context passport is maintained by an active user agent which communicates with the services via a specific protocol.

A flexible protocol is required for this communication between context passport and the service-specific personalization component. Such a protocol has to support the negotiation of the user model information to be exchanged and the bidirectional exchange of user information. In more detail such a protocol operates in three phases a) negotiation phase, b) personalization phase, and c) synchronization phase. In the negotiation phase, the Context Passport and the service agree on information to be exchanged. The main goal to be achieved is a common understanding on the type of information that the other partner can understand and use. In our approach the UUCM provides the common vocabulary for negotiating about the available user information (dimensions, facets about the user, etc.) In order to perform an automatic negotiation about what activities can be supported, there needs to be an agreement on a machine understandable common vocabulary or ontology of the respective domain (e.g. Travel). After an agreement has been reached on the activity to be performed and the available user information, the Context Passport needs to extract information relevant to this activity (context selection). This is communicated to the system in the personalization phase. After the personalized activity has been performed, the respective service has a slightly changed understanding of the user. In the synchronization phase the service informs the context passpoprt about these changes keeping the Context Passport uptodate.

There is thus a requirement from bidirectional information exchange so that other services may benefit from up-to-date information about the user. An early implementation of the Context Passport has been done in the WWW scenario, which supports web systems for Cross-system Personalization. This implementation supports an early version of the CSCP enabling synchronization of user profiles between two test web systems. Implementation details are available in [20]. Future implementations will support task based reasoning and relationship based recommendations.

4 Related Work

Metadata Management Decentralized and peer-to-peer systems can be considered as a further generalization of distributed systems. Therefore, decentralized data management has much in common with distributed databases, which are already well explored [21, 22]. However, some important differences exist. Distributed databases are made to work in stable, well connected environments (e.g. LANs) with the global system overview, where every crashed node is eventually replaced by a new proper one. Also, they need some sort of administration and maintenance.

On the contrary, the P2P systems are deployed mostly on the highly unreliable Internet. Some links can be down, network bandwidths are not guaranteed. The P2P systems allow disconnection of any peer at any time, without a need for replacement, and none of the peers is aware of the complete system architecture. Therefore, the system must self-organize in order to survive such situations.

Many distributed databases like Teradata, Tandems NonStopSQL, Informix Online Xps, Oracle Parallel Server and IBM DB2 Parallel Edition [23] are available on the market. The first successful distributed filesystem was Network File System (NFS) succeeded by Andrew File System (AFS), Coda and xFS, etc.

Current popular P2P file-sharing systems (e.g. KaZaA, Gnutella, eDonkey, Past [3]) might be a good starting point for enabling decentralized data management. However, these systems have some important drawbacks: file-level granularity and write-once access, i.e. files are non-updateable after storing. Storing a new version requires a new filename. Usually, a file contains many objects. As a consequence, retrieving a specific object would require getting the whole file first. If a object must be updated, then a whole new file version must be created and stored. In current systems it is not possible to search for a particular object inside the files. The query results contain the whole files, not only requested objects. Advanced searching mechanism like qualified, range or boolean predicates search is not supported. Usually, metadata have rich and complex structure and queries on them are more than simple keyword match. Also, metadata should be updateable. Thus, the presented P2P systems are not suitable for decentralized metadata management.

There are some attempts [24] to extend Gnutella protocols to support other types of queries. It would be quite possible to create a Gnutella implementation that understands some variant of SQL, XPath or XQuery. However, such networks would have problems with system load, scalability and data consistency, e.g. only locally stored data could be updated and mechanisms for updating other replicas do not exist.

Information Retrieval Typical Peer-to-peer information retrieval (P2PIR) methods are working decentralized, as proposed by the P2P paradigm [3]. No server is involved as it would be in a hybrid or client-server architecture. Common P2PIR approaches let the requesting peer contact other peers in the network for the desired documents. In the worst case, the query is broadcast to the whole network resulting in lots of communication overhead. Another approach would be to store all index information on every peer and search for relevant documents locally. Peers would request the required information during the inital introduction phase, and updates would be spread from time to time. However, this approach is not feasible since the expected storage costs would be quite high. Intermediate approaches which try to balance communication and storage costs work with peer content representations like the clustering approach discussed in [25]. Such a peer content representation does not need the amount of data a snapshot of the whole distributed index would need, but conveys enough information to estimate the probability that a documents relevant to the query can be found on a certain peer.

Some annotation systems [26] provide simple full-text search mechanisms on annotations. The Yawas system [27] offers some means to use annotations for document search, e.g. by enabling users to search for a specific document type considering annotations. Golovchinsky *et al.* [28] use annotations as markings given by users who judge certain parts of a document as being important when emphasizing them. Their approach gained better results than classic relevance feedback, as experiments showed. Agosti *et* *al.* [11] discuss facets of annotations and propose an annotation-based retrieval function based on probabilistic inference. Frommholz *et al.* [29] present a nested annotation retrieval approach based on probabilistic logics. This approach does not only consider syntax and semantic of annotations, but makes use of (explicitly given) discourse structure relations among them. The idea of automatic annotations is motivated by the internal linking methods described in [15] by Thiel *et al*. Related to this is the overview given by Agosti and Melucci in [30], where they discuss how to use informatio retrieval techniques to automatically create hypertexts.

Personalization Support The most popular personalization approaches in digital libraries or more general in information and content management systems are recommender systems and methods that can be summarized under the term personalized information access. Recommender systems (see e.g. [31]) give individual recommendations for information objects following an information push approach, whereas personalized information access (personalized newspapers, etc.) is realized as part of the information pull process, e.g. by filtering retrieval results or refining the queries themselves.

Personalization methods are based on modeling user characteristics, mainly cognitive pattern like user interests, skills and preferences [32]. More advanced user models also take into account user tasks [33] based on the assumption that the goals of users influence their needs. Such extended models are also referred to as user context models [34]. A flexible user context model that is able to capture an extensible set of user model facets as it is required for cross-service personalization can be found in [19]. Information for the user models (also called user profiles) are collected explicitly or implicitly [35], typically by tracking user behavior. These user profiles are used for personalized filtering in information dissemination (push) as well as in information access (pull) services. An important application area is personalized information retrieval. The information about the user is used for query rewriting [36], for the filtering of query results [37] as well as for a personalized ranking of query results [38].

5 Conclusions and Future Work

In this paper, we discussed opportunities and challenges for information access support resulting from the transition of more traditional, centrally controlled DL architectures to DLs as dynamic federations of content collections and DL services.

The discussion focussed on metadata management, information retrieval, and personalization support. In addition to discussing the central challenges, an advanced approach has been discussed for each of the three aspects: For metadata management a decentralized P2P data store solves the problem of systematic and efficient decentralized metadata management. Applications of annotations and annotation-based retrieval in the P2P context is considerd as a way to improved information retrival support in a decentralized environment. Finally, cross-service personalization is discussed as an adequate way to handle personalization in a dynamic service-oriented environment.

The list of the considered information access issues discussed is not meant to be exhaustive. Further challenges raise within next-generation DL architectures like effective metadata brokering and advanced methods for ensuring content security and quality. The envisaged support for information access needs to combine

the approaches mentioned above in a balanced way to ensure
that users will benefit from decentralized architectures, while
at the same time, maintaining the high level of organization and reachability
that users of DL systems are used to.

Such issues are addressed in the BRICKS and the DIIGENT project in which our institute is involved together with partners from other European countries.

References

1. BRICKS Consortium: BRICKS - Building Resources for Integrated Cultural Knowledge Services (IST 507457). (2004) `http://www.brickscommunity.org/`.
2. DILIGENT Consortium: DILIGENT - A DIgital Library Infrastructure on Grid ENabled Technology (IST 004260). (2004) `http://www.diligentproject.org/`.
3. Milojičić, D., Kalogeraki, V., Lukose, R., Nagaraja, K., Pruyne, J., Richard, B., Rollins, S., Xu, Z.: Peer-to-peer computing. Technical report (2002) `http://www.hpl.hp.com/techreports/2002/HPL-2002-57.pdf`.
4. Rowstron, A., Druschel, P.: Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. Lecture Notes in Computer Science **2218** (2001)
5. Aberer, K.: P-Grid: A self-organizing access structure for P2Pinformation systems. Lecture Notes in Computer Science **2172** (2001)
6. Stoica, I., Morris, R., Karger, D., Kaashoek, F., Balakrishnan, H.: Chord: A scalable Peer-To-Peer lookup service for internet applications. In: Proceedings of the 2001 ACM SIGCOMM Conference. (2001) 149–160
7. Knežević, P.: Towards a reliable peer-to-peer xml database. In Lindner, W., Perego, A., eds.: Proceedings ICDE/EDBT Joint PhD Workshop 2004, P.O. Box 1527, 71110 Heraklion, Crete, Greece, Crete University Press (2004) 41–50
8. W3C: Web Services Description Language (WSDL) 1.1. (2001) `http://www.w3.org/TR/wsdl`.
9. OASIS: Universal Description, Discovery and Integration (UDDI). (2001) `http://www.uddi.org/`.
10. Risse, T., Knežević, P.: Data storage requirements for the service oriented computing. In: SAINT 2003 - Workshop on Service Oriented Computing. (2003) 67–72
11. Agosti, M., Ferro, N., Frommholz, I., Thiel, U.: Annotations in digital libraries and collaboratories – facets, models and usage. In: Proc. 8th European Conference on Research and Advanced Technology for Digital Libraries (ECDL). (2004) 244–255
12. Searle, J.: A taxonomy of illocutionary acts. In Searle, J., ed.: Expression and Meaning. Studies in the Theory of Speech Acts. Cambridge University Press,, Cambridge (1979) 1–29
13. Frommholz, I., Brocks, H., Thiel, U., Neuhold, E., Iannone, L., Semeraro, G., Berardi, M., Ceci, M.: Document-centered collaboration for scholars in the humanities - the COLLATE system. [39] 434–445
14. Agosti, M., Ferro, N.: Annotations: Enriching a Digital Library. [39] 88–100
15. Thiel, U., Everts, A., Lutes, B., Nicolaides, M., Tzeras, K.: Convergent software technologies: The challenge of digital libraries. In: Proceedings of the 1st Conference on Digital Libraries: The Present and Future in Digital Libraries, Seoul, Korea (1998) 13–30
16. Golovchinsky, G.: What the query told the link: The integration of hypertext and information retrieval. In Bernstein, M., Osterbye, K., Carr, L., eds.: Proceedings of the 8th ACM Conference on Hypertext (Hypertext '97), Southampton, UK, ACM Press, New York, USA (1997) 67–74

17. Attardi, G., Gullí, A., Sebastiani, F.: Automatic Web page categorization by link and context analysis. In Hutchison, C., Lanzarone, G., eds.: Proceedings of THAI-99, 1st European Symposium on Telematics, Hypermedia and Artificial Intelligence, Varese, IT (1999) 105–119
18. Neuhold, E.J., Niederée, C., Stewart, A.: Personalization in digital libraries: An extended view. In: Proceedings of ICADL 2003. (2003) 1–16
19. Niederée, C., Stewart, A., Mehta, B., Hemmje, M.: A multi-dimensional, unified user model for cross-system personalization. In: Proceedings of Advanced Visual Interfaces International Working Conference (AVI 2004) - Workshop on Environments for Personalized Information Access, Gallipoli (Lecce), Italy, May 2004. (2004)
20. Mehta, B., Niederée, C., Stewart, A.: Towards cross-system personalization. In: To appear in Proceedings of UAHCI 2005, Las Vegas, Nevada, July 2005. (2005)
21. Özsu, M.T., Valduriez, P.: Principles of Distributed Database Systems. Prentice Hall (1999)
22. Bernstein, P.A., Hadzilacos, V., Goodman, N.: Concurency Control and Recovery in Database Systems. Addison-Wesley (1997)
23. Brunie, L., Kosch, H.: A communications-oriented methodology for load balancing in parallel relational query processing. In: Advances in Parallel Computing, ParCo Conferences, Gent, Belgium. (1995)
24. GPU: A gnutella processing unit (2004) `http://gpu.sf.net`.
25. Müller, W., Henrich, A.: Fast retrieval of high-dimensional feature vectors in P2P networks using compact peer data summaries. In: Proceedings of the 5th ACM SIGMM international workshop on Multimedia information retrieval, ACM Press (2003) 79–86
26. Ovsiannikov, I.A., Arbib, M.A., McNeill, T.H.: Annotation technology. Int. J. Hum.-Comput. Stud. **50** (1999) 329–362
27. Denoue, L., Vignollet, L.: An annotation tool for web browsers and its applications to information retrieval. In: Proceedings of RIAO 2000, Paris, April 2000. (2000)
28. Golovchinsky, G., Price, M.N., Schilit, B.N.: From reading to retrieval: Freeform ink annotations as queries. In Gey, F., Hearst, M., Tong, R., eds.: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, New York, ACM Press (1999) 19–25
29. Frommholz, I., Thiel, U., Kamps, T.: Annotation-based document retrieval with four-valued probabilistic datalog. In Roelleke, T., de Vries, A.P., eds.: Proceedings of the first SIGIR Workshop on the Integration of Information Retrieval and Databases (WIRD'04), Sheffield, UK (2004) 31–38
30. Agosti, M., Melucci, M.: Information retrieval techniques for the automatic construction of hypertext. In Kent, A., Hall, C.M., eds.: Encyclopedia of Library and Information Science. Volume 66. Marcel Dekker, New York, USA (2000) 129–172
31. Bouthors, V., Dedieu, O.: Pharos, a collaborative infrastructure for web knowledge sharing. In Abiteboul, S., Vercoustre, A.M., eds.: Research and Advanced Technology for Digital Libraries, Proceedings of the Third European Conference, ECDL'99, Paris, France, September 1999. Volume LNCS 1696 of Lecture Notes in Computer Science., Springer-Verlag (1999) 215 ff.
32. McTear, M.: User modeling for adaptive computer systems: A survey of recent developments. In: Artificial Intelligence Review. Volume 7. (1993) 157–184
33. Kaplan, C., Fenwick, J., Chen, J.: Adaptive hypertext navigation based on user goals and context. In: User Modeling and User-Adapted Interaction 3. Kluwer Academic Publishers, The Netherlands (1993) 193–220
34. Goker, A., Myrhaug, H.: User context and personalization. In: Proceedings of the European Conference on Case Based Reasoning (ECCBR 2002) - Workshop on Personalized Case-Based Reasoning, Aberdeen, Scotland, 4-7 September 2002. Volume LNCS 2416 of Lecture Notes in Artificial Intelligence., Springer-Verlag (2002)

35. Pretschner, A., Gauch, S.: Personalization on the web. Technical Report ITTC-FY2000-TR-13591-01, Information and Telecommunication Technology Center (ITTC), The University of Kansas, Lawrence, KS (1999)
36. Gulla, J.A., van der Vos, B., Thiel, U.: An abductive, linguistic approach to model retrieval. Data & Knowledge Engineering **23** (1997) 17–31
37. Casasola, E.: Profusion personalassistant: An agent for personalized information filtering on the www. Master's thesis, The University of Kansas, Lawrence, KS (1998)
38. Meng, X., Chen, Z.: Personalize web search using information on client's side. In: Proceedings of the Fifth International Conference of Young Computer Scientists, August 17-20, 1999, Nanjing, P.R.China, International Academic Publishers (1999) 985–992
39. Koch, T., Sølvberg, I.T., eds.: Proc. 7th European Conference on Research and Advanced Technology for Digital Libraries (ECDL). In Koch, T., Sølvberg, I.T., eds.: Proc. 7th European Conference on Research and Advanced Technology for Digital Libraries (ECDL), Lecture Notes in Computer Science (LNCS) 2769, Springer, Heidelberg, Germany (2003)