Integrating Multi-dimensional Information Spaces

Kostas Saidis and Alex Delis
{saiko,ad}@di.uoa.gr

Department of Informatics and Telecommunications
University of Athens, 157 84, Athens, Greece

1 Introduction

Contemporary digital libraries (DLs) and related systems, including but not limited to institutional repositories, e-publishing platforms and cultural heritage systems, face the need to manage diversely structured information spaces. The latter are comprised of multi-faceted digitized and/or born-digital content, such as intellectual works, scholarly information, institutional or personal archives, cultural heritage material and even user-generated content such as blogs and video casts. Although such information spaces may –and in practice, actually do– originate from heterogeneous sources, including XML repositories or custom database solutions, the increased connectivity offered by the expansion of the Internet, raises the challenge to inter-connect such heterogeneous sources and integrate their information spaces in order to enable users to share, reuse, refine and extend them in varying application contexts.

In this paper, we view VLDLs as DLs that manage "large information spaces", not only in terms of the volume of content they can manage but also in terms of the diversity of material and the heterogeneity of sources they can support. Under this perspective, the information integration/interoperation requirements mentioned above are of significant importance to VLDL development. As we discuss herein, *VLDLs are dominated by the multi-dimensional nature of information space management*, as comprised of information discovery, access, conceptualization and utilization options. In particular, the ability to extend information spaces' management options in the aforementioned dimensions is strongly connected to the ability to integrate these spaces. Based on this, we identify several essential design issues of a common information space management infrastructure. This infrastructure will add value to VLDLs by simplifying –and automating as much as possible– complex information space integration tasks.

2 Rationale & Motivating Example

Making systems interoperate and integrate their information spaces is hard, because, in practical terms, different systems develop different views of digital content for different purposes, as Figure 1 shows. Viewing information space integration as a process roughly comprised of the following steps:

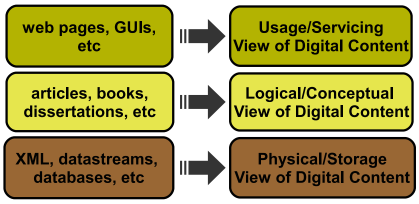
web pages, GUIs, etc
Usage/Servicing View of Digital Content
articles, books, dissertations, etc
Logical/Conceptual View of Digital Content
XML, datastreams, databases, etc
Physical/Storage View of Digital Content

Fig. 1. Multiple views of digital content

a. information discovery: systems discover or "learn about" the existence of each other.

b. information identification/access: systems can unambiguously identify individual elements of their information spaces, while they are also provided with a means to access such elements.

c. information utilization: systems can synthesize the elements of their information spaces to accommodate to their particular service provision requirements.

it is rather clear that to perform the above steps requires dealing with a plethora of information discovery, access, conceptualization and utilization options supported by involved systems. Indeed, due to the multi-dimensional nature of digital content management, each system supports its own *information access* options to render its own *information conceptualizations* required to answer its particular *information utilization and usage* needs. Thus, when integrating information spaces, we practically need to extend involved systems in terms of multiple information management dimensions to:

- support new information discovery mechanisms,
- access new kinds of information sources,
- augment the system with new content usage scenarios
- revisit existing information conceptualizations

Realizing such interdependent and crosscutting extensions can be as complex and costly as adding a brand new set of features to a system, often "breaking" its existing design. However, cost-effectiveness is crucial for information integration/interoperability, since the latter is about "*enabling information that originates in one context to be used in another in ways that are as highly automated as possible*" [6]. Therefore, the success of an approach to integrate information spaces depends highly on *the level of automation achieved*, as an approach which involves the manual execution of costly and complex tasks is most likely to be ineffective in practice.

The Web, for example, fully automates information identification and access through the use of URLs and HTTP respectively. Information discovery is automated by value-added search services such as Google, while the Web's information utilization options, although automated, are based on a rather limited "document-based" conceptualization that adheres to a publish/consume paradigm. Technologies such as Web Services [8] and the Semantic Web [5] aim at enhancing the limited discovery and utilization options offered by the traditional Web. On one hand, Web Services can be used to discover and utilize "live" content comprised of both data and code. On the other, the Semantic Web extends the "document-based" nature of the traditional Web with the addition of semantic annotations to Web resources. Such annotations can then be used to enhance Web resource utilization in various contexts.

Although the Web is the world's largest interoperable information space, similar, if not identical, information discovery/identification, access, conceptualization and utilization issues arise in integrating smaller-scale information spaces. For example, our Pergamos DL offers a uniform platform for documenting, preserving and publishing the digital material of University of Athens [4]. Pergamos has been in production use for more than 4 years, currently hosting about 300,000 items exceeding 1 TB of space. These items are grouped in various collections that originate from digitization projects of various University departments, while upon their completion, collections are published through Pergamos' web-based front-end [7]. However, Pergamos is not the only DL system available at the University, as many departments have deployed various assorted DLs over the years, comprised of diverse collections held in heterogeneous storage solutions. For instance, there is *Anthemion*, a client-server database application hosting digitized books of the early 1900's. There is also a Lotus Domino application hosting theses and dissertations of the Faculty of Law. In this context we face the following dual challenge. On one hand we need to extend Pergamos to support new digital collections, as new digitization projects emerge. On the other, we need to make Pergamos interoperate with *Anthemion* and the Domino-based "legacy" DLs.

Any successful system will, sooner or later, need to extend its multiple information space management options. For example, Pergamos has to access Domino-based dissertations and Anthemion's database-oriented books in order to publish them using its existing web-based front-end. To realize this requires supporting new kinds of (a) information sources, extending Pergamos' information access options, (b) content items, namely, books and dissertations, thus extending Pergamos' information conceptualization options.

Moreover, VLDLs should share but also reuse, refine and/or extend their information spaces. For instance, in Pergamos we need to integrate information spaces that goes beyond sharing read-only digital content representations. In particular, we need Pergamos back-end system, used by our libraries staff to document content items through web-based forms, to be able to access, yet also update and modify the Domino and Anthemion collections. This yields an additional need to extend existing content utilization options, as Pergamos back-end should be made to update these newly added collections, not only access them in a read-only fashion. Finally, Pergamos' information discovery options should be also extended in order to realize cross-DL searching and retrieval.

This highlights the challenges that arise in contemporary DLs and related systems caused by the need to amalgamate their information spaces. This need has a drastic affect on DL systems, as it imposes extensions in multiple of their information space management options. Therefore, it is rather critical to supply systems with an infrastructure for performing such extensions in a straightforward and cost-effective fashion.

3 Multi-dimensional Information Space Management

Figure 2 shows a VLDL managing a multi-dimensional information space. The former consists of the services, which, instrumented by the VLDL's application logic, synthesize the information space in terms of various information access, conceptualization, utilization and discovery options.

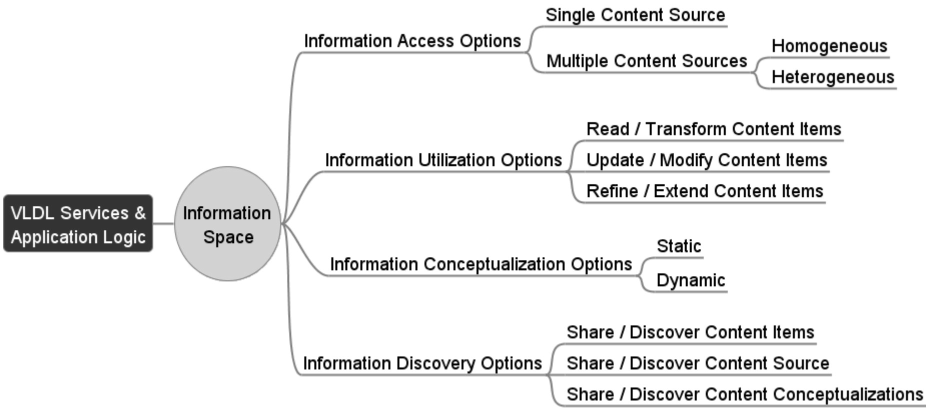
VLDL Services & Application Logic
Information Space
Information Access Options
Single Content Source
Multiple Content Sources
Homogeneous
Heterogeneous
Information Utilization Options
Read / Transform Content Items
Update / Modify Content Items
Refine / Extend Content Items
Information Conceptualization Options
Static
Dynamic
Information Discovery Options
Share / Discover Content Items
Share / Discover Content Source
Share / Discover Content Conceptualizations

Fig. 2. Multi-dimensional Information Space Management Options

The figure can be viewed as a taxonomy of a DL's information space management features, identifying several essential characteristics of its overall functionality. For example, in a metadata harvesting scenario, a central DL employs a harvesting protocol to gather scholarly information from various sources. Although individual content sources may be heterogeneous, the use of a common high-level harvesting protocol hides such details from the central DL, thus enabling the latter to treat all sources in a unified manner, as if they were homogeneous, advancing interoperability. Moreover, given that this scenario involves scholarly material, the central DL can issue a static information conceptualization of "article objects".

Clearly, the information space management options of Figure 2 are highly coupled to each other, reflecting the complexity of interoperable DLs. In a technical context, the addition of value offered by the use of common information space management infrastructure resides in enabling DL application developers and designers to treat the four information space management dimensions of Figure 2 independently of each other. This will offer standard and effective mechanisms to extend a DL's information discovery, access, utilization and conceptualization options, ideally without (a) affecting the functionality of the DL system in the remainder dimensions, (b) requiring any complex and costly DL system code modifications and (c) causing a DL shutdown and/or restart. Placing the infrastructure between a DL's application logic and its information space, in the remainder of this section we identify some essential design issues that help achieve the aforementioned goal.

3.1 Information Access Options

We use the information access dimension to render a logical view of a system's content access capabilities, reflecting *the fundamental functionality to contact a content source and fetch its items.* The system will employ a specific "data access machinery" to contact a content source, a set of access mechanisms and related tools, including involved protocols, XML handling libraries or database access components. From a system design perspective, we view sources that are accessed using a common "data access machinery" as homogeneous and those which require different machineries as heterogeneous. We identify the following three major information access options, where a system may operate atop:

1. *a single content source*: all the operations related to information storage and retrieval are performed via a single source. For example, a personal DL is expected to employ a local content store solution, while a web-based DL may use a more sophisticated multi-tier architecture. Yet, in both cases, the DL system in question uses a single "data access machinery".
2. *multiple homogeneous sources*: the system employs several homogeneous sources to meet its application logic requirements. One such example is the metadata harvesting system described before. Another example may be a load balancing scenario, where a DL application server redirects access requests to a set of identical content sources based on the system's load. In both examples, the content sources are homogeneous, as the DL system in question uses a single "data access machinery" to contact all of them in a unified manner.
3. *multiple heterogeneous sources*: the DL system has to combine the usage of various "data access machineries" to contact a variety of content sources, in order to answer the needs of its application logic. An example of such a case is our Pergamos front-end, which has to manage content that originates from the Pergamos XML repository, the Domino-based theses and dissertations and Anthemion's custom database.

The infrastructure should enable DLs to operate atop multiple heterogeneous sources in a cost-effective manner. For example, should we consider a DL that supports a single content source, the challenge here is to progressively extend its access options to include more sources, as new content access needs emerge. To this end, the infrastructure should offer a unified, system-wide *Content Access API*, used to (a) register new sources in the information space, dealing with their specific "data access machineries" and (b) allow DL services to fetch underlying content items uniformly, regardless of their location and storage details.

3.2 Information Utilization Options

We use the information utilization dimension to refer to the essential operations a system performs on its content items, namely:

1. *read/transform content items*: this is a fundamental operation, which, as mentioned before, is offered by the unified *Content Access API*, allowing DL services to to fetch any content item originating from any content source. Services can then synthesize the data held in the item to meet any read-only service provision requirement, such as dissemination or web publishing.
2. *update/modify content items*: except of the ability to fetch content items, DL services may also need to update/modify content items. The infrastructure should answer such a need by offering an appropriate *Content Update API*, providing a unified, system-wide support for performing update/modify operations on the data held in the content items. For example, our Pergamos back-end cataloging UI should use this API to generate web forms for allowing catalogers to create, document, update and modify content items. The role of this API is to assist DL implementors to realize high-level information update services, such as cataloging web forms, administration UIs, ingestion or migration services, while maintaining the ability to operate atop multiple heterogeneous sources.
3. *refine/extend content items*: this is the most advanced case, as it refers to the ability to modify the structure of content items, not only the data they hold. Ideally, the ability to build upon existing content items to generate new ones should be also given. Content item refinement and extension is strongly coupled to the conceptualization options supported by the DL, as discussed in the next section.

3.3 Information Conceptualization Options

Consider a system such as *Anthemion* which manages “book” items. Although we may all share a common perception of the “book” concept, in a technical context, “book” items may have different conceptualizations in different systems. For example, in a context where books are digitized, it is not uncommon to represent such “books” as compound entities comprised of “pages”. In another context, where a book’s content is born-digital, as in the cases of e-books or other similar web material, it is rather common to represent “books” in a more simple fashion. Finally, in a scholarly information context, a conference’s proceedings may be realized in terms of “book” items comprised of “articles”.

Since DLs issue various conceptualizations for their content items, we use the information conceptualization dimension to render a high-level view of a system’s conceptualization capabilities, where a system supports:

1. *static information conceptualizations*: *Anthemion*’s information space is comprised of “book” items, while the Domino application manages “dissertation” items. The conceptualizations supported by these DLs are *static*, in the sense that they are hard-coded in the DL application code. Should

we consider that a need to support some new types of content items occurs, we cannot make such systems answer this need without performing costly modifications to their source code. The same stands when a need to modify the structure of existing "book" and/or "dissertation" items occurs.

2. *dynamic information conceptualizations*: the infrastructure should follow a more *laissez faire* approach on expressing and handling information conceptualizations, as ideally, it should be able to support diverse content conceptualizations dynamically. Viewing content conceptualizations as definitions of the structure and relationships of content items, the infrastructure should offer the means to support varying kinds of such definitions, which can be added into the information space in a dynamic fashion. DL services could then use these definitions to identify the conceptualizations pertinent in the information space and adjust to their requirements dynamically. This way, the system will also adapt more effectively when the information space needs to be augmented with a new type of content items.

   Moreover, given that conceptualization definitions will be independent of any particular content storage and content utilization details, the infrastructure could also employ an inheritance mechanism for building upon existing definitions to generate new ones, allowing users and services to refine and extend conceptualizations in various information spaces.

3.4 Information Discovery Options

In terms of information discovery, it is a commodity for a system to employ a search utility to assist users –or other systems– to search its information space and retrieve items that match specific criteria. Search functionality can be primarily viewed as a helper for simplifying information space usage, yet, it is also coupled to the information access options supported by a system, as indexing facilities usually accompany data storage systems. For example, in a scenario where various content sources participate in a DL federation, one may use a centralized indexing approach where all content is indexed in a central indexing facility. Another approach could distribute search queries to the individual indexing facilities that may be present in the network.

In the scope of this paper we place our focus on identifying the information that should be possible to share/discover using the infrastructure, namely:

1. *share/discover content items*: this refers to the main information discovery requirement, the ability to search for specific content items.
2. *share/discover content sources*: the infrastructure should also provide the ability to share and discover content sources. This will allow remote DLs to gain access to their underlying content sources, further fostering their integration. For example, a DL may publish an index of its content sources along with a wrapper of its *Content Access API* –e.g., a web service. This will allow remote systems to reuse the DL's information access options and contact the DL's sources to access hosted items.
3. *share/discover content conceptualizations*: the ability to share/discover definitions of content conceptualizations should be provided, as it will enable

DLs to discover new kinds of content items. For example, in a fashion similar to the above content source sharing scenario, a DL may publish its content conceptualization definitions, offering this way access to the structure and relationships of its content items. This will enable remote DLs to use these definitions in various usage scenarios –e.g., to build upon existing definitions to generate new conceptualizations, or to use the definitions to generate a content item browsing service over the whole DL network.

Thus, the infrastructure should need to provide three information discovery facilities: one for searching content items, one for discovering content sources and one for sharing content conceptualizations.

4 Closing Remark

In this paper, we have identified several essential requirements for supplying DLs with a common information space management infrastructure. Firstly, the infrastructure should treat information spaces as multi-dimensional information management contexts, comprised of information discovery, access, conceptualization and utilization options. Secondly, in order to simplify integration and interoperation of information spaces, it should automate the extension of these options independently of each other. We believe that such an approach can assist the goal of the DL research community, as expressed in efforts such as the DELOS reference model [1], the 5S model [3] and the OAIS reference model [2], to offer a unified DL foundation.

References

1. L. Candela, D. Castelli, P. Pagano, C. Thanos, Y. Ioannidis, G. Koutrika, S. Ross, H.-J. Schek, and H. Schuldt. Setting the Foundations of Digital Libraries: The DELOS Manifesto. *D-Lib Magazine*, 13(3/4), March/April 2007. [doi:10.1045/march2007-castelli].
2. Consultative Committee for Space Data Systems (CCSDS). Reference Model for an Open Archival Information System (OAIS). Blue Book, Issue 1, http://public.ccsds.org/publications/archive/650x0b1.pdf.
3. M. Gonçalves, E. Fox, L. Watson, and N. Kipp. Streams, Structures, Spaces, Scenarios, Societies (5s): A Formal Model for Digital Libraries. *ACM Transactions on Information Systems (TOIS)*, 22(2):270–312, 2004.
4. K. Saidis, G. Pyrounakis, M. Nikolaidou, and A. Delis. Digital object prototypes: An effective realization of digital object types. In *ECDL '06: Proceedings of the 10th European Conference on Digital Libraries*, Alicante, Spain, September 2006.
5. N. Shadbolt, T. Berners-Lee, and W. Hall. The Semantic Web Revisited. *IEEE Intelligent Systems*, 21(3):96–101, 2006.
6. The International DOI Foundation. The DOI Handbook. Edition 4.4.1, October 2006, [doi:10.1000/182].
7. University of Athens. Pergamos Digital Library. http://pergamos.lib.uoa.gr/.
8. *Web Services Activity.* World Wide Web Consortium. Available at http://www.www.org/2002/ws/.