A CONCEPTUAL MODEL FOR DIGITAL LIBRARIES EVOLUTION

Keywords: Digital libraries, Metadata, Service-Oriented Architecture, Multi-Agent Systems, Schema Evolution.

Abstract: The evolution and preservation of digital libraries are not simply a matter of technological decisions, but they can be better understood if treated as the integration of three complementary dimensions (based on the informational, technological and social domains). These dimensions together form a conceptual framework suitable to characterize the whole digital library concept.
In this paper, starting from the experience and the lessons learned in the realization of the EU-India E-Dvara project, we propose such framework, providing motivational examples and discussing opportune solutions. More in particular, we discuss the issues concerned the technical infrastructure adaptation, the coordination of different user roles, and the data evolution in order to select the dimensions along which we base our framework.

1 INTRODUCTION

Many works coming from both the academia and the industry seem to suggest that preservation and evolution of digital libraries are firstly a matter of technological issues (e.g. (Barkstrom et al., 2002)). We certainly recognize the need of data storage infrastructures, knowledge management systems (metadata and search mechanisms) or data transport and security facilities. However, the technology should be viewed "simply" as a means to provide the services typically built around a digital archive. We recognize a deeper meaning in the evolution phenomena of digital libraries, taking into account also social aspects such as the diverse range of actor roles involved in the content production and exploitation processes. Thus, we contrast the "technology-centered" vision, characterizing the evolution of both the digital content and the services built upon it as the integration of three complementary dimensions (social, technological and informational) which form together a conceptual framework suitable to better formalize the digital library concept and its evolution issues over the time.

This paper is based on a three-years experimentation with the EU-India E-Dvara project[1]: a digital platform devoted to e-content management in Indian heritage and sciences. We have already published three other works on the subject, discussing the overall project goals, and the technical details concerning both the data representation model and the general software architecture.

The main and new contribution of this work is the characterization of a digital library according to its evolution aspects; in particular, after a brief survey dedicated to related works, we:

- introduce a conceptual framework to handle the evolution of digital archives along multiple dimensions (Section 3);
- provide representative examples concerning evolution issues, weaknesses, and mistakes emerged during the evaluation of our current E-Dvara prototype (Section 4.1 - Section 4.3);
- propose a new, distributed approach to handle evolution open problems (Section 5).

[1] http://edvara.uniud.it/india

2 RELATED WORKS

In the last few years several research projects have been proposed in order to cope the same requirements (concerning data preservation and organization) we are currently facing. For example, the storage of XML-based document, which is one of the core architectural properties of e-Dvara, has been previously proposed in Greenstone (Bainbridge et al., 2001; Witten et al., 2000), a digital library designed to provide librarians the ability to create and publish heterogeneous collections of digital contents on the Web like text, images, videos and e-books. Each content in Greenstone can be described using *metadata*, either imported from standard schemas (e.g. Dublin Core[2]) or manually provided by librarians. However, Greenstone does not provide any policy or roles for the management of the content submission process. Moreover, it does not provide functionalities concerning the evolution management of both contents and collection templates.

D-Space (Tansley et al., 2003) is an author-oriented distributed digital library aimed at providing long-term preservation of heterogeneous contents, by improving some of the limitations affecting Greenstone. It provides long-term preservation facilities, by assigning a persistent identifier to each submitted resource and supporting software and hardware methodologies for data backup and content versioning. D-Space introduces also a multi-roles approach to content publishing, identifying the following actors: (1) authors and organizations, providing the contents, (2) librarians, performing content validation, and (3) users, interested in content retrieval. Content-based workflows can be customized in order to cope with the needs of specific organizations. Part of the policies defined in D-Space have been introduced also in E-Dvara to structure content and to delegate proper activities to different stakeholders.

Service-oriented architecture and data interoperability issues in digital libraries have been explored also by the Fedora Project (Lagoze et al., 2005), a distributed architecture for contents publishing, aggregation and retrieval. Composite information is obtained by means of aggregation of physical contents, viewed as bit-streams, located worldwide into the Fedora repositories. Preservation of each content is achieved by means of a naming service, which can be used to access the selected content. In addiction to composition, Fedora provides users the ability to define new contents by applying to existing physical objects custom components called disseminators (e.g.: a thumbnails generator applied to high-resolution pictures or videos). Both Fedora and E-Dvara allow content editors and archivists to define semantic connections between archived contents. In Fedora, however, connections are defined between two contents treated as set of physical contents. E-Dvara, vice versa, allows content writers to define relations implementing a specific template which enhances a closer *semantic validation* of the content.

Other approaches in designing digital libraries for content preservation are described in (Bekaert et al., 2005; Lutzenkirchen, 2002); the aDORe project, in particular, adopts the MPEG-21 DID content representation model in order to provide preservation and retrieval of heterogeneous multimedia contents.

The above mentioned systems are centered on contents, defined as *binary resources* enriched by metadata devoted to preservation, storage and retrieval purposes, but not intended for data structuring, as we do in E-Dvara. Thus, preservation and evolution of a data model is implemented as a low-level mechanism, where data is processed as bit-streams instead of as instances of well-defined structures (i.e. XML Schema). In E-Dvara we provide preservation facilities, managing both physical and *logical evolution* of the stored data. More specifically, E-Dvara is conceived to explicitly deal with evolution of a data model by means of preserving *information integrity*.

Finally, in this work we extend the digital library model proposed by Yates (Yates, 1989) and later refined by Rowlands-Bawden (Rowlands and Bawden, 1999). We characterize the evolution of archived information by additing the concept of *evolution dimension*.

3 THE CONCEPTUAL FRAMEWORK

Adopting a conceptual framework is an important step to better understand the evolution of a digital library, especially for bridging the gap between research and practice (Bawden and Rowlands, 1999). For this purpose, we start from the topology provided by Yates (Yates, 1989), incorporating the vocabulary suggested by Rowlands-Bawden (Rowlands and Bawden, 1999). Then, we describe the evolution of each concept (point) in the topology by consider the different directions from which we can reach it. The result is the extended conceptual framework illustrated in Figure 1, which highlights three domains:

- The *Informational domain*, which describes knowledge organization and description (e.g. metadata).

[2]See for more details: http://dublincore.org/

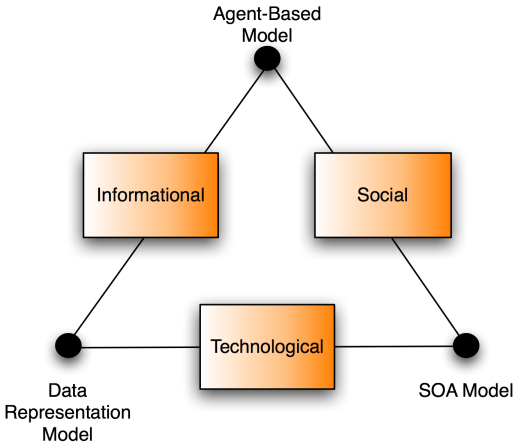

Figure 1: The evolution conceptual model

- The *Technological domain*, which describes knowledge organization and discovery (e.g. software agents), technical impacts on the information transfer chain, technology factors (e.g. human-computer interaction).
- The *Social domain*, which describes human and organizational factors, information laws and policies, social impacts on the information transfer chain, and library management concerns.

We define a specific evolution dimension combining together every pair of domains. Moreover, we believe that the Yates model, which forms the core of our framework[3], is still actual nowadays; indeed, a (digital) library stores (and provides access to) documents, which are created and maintained using technologies, and both documents and technologies are deployed to support the work of librarians, researchers, and readers (the final users of a library).

The open issues faced during our experimentation with E-Dvara may be identified along three evolution dimensions:
1. *Informational-Technological* dimension, which identifies all *data evolution* problems due to changes in the underlying data model (data schema); they invalidate entire archives of documents conforming to the old schema version.
2. *Technological-Social* dimension, which identifies problems concerning the need to adapt the technical infrastructure of a digital library in order to fulfill new user requirements (e.g. the integration of heterogeneous services to support the interaction with different new user roles, such as virtual museum curators, Web 2.0 communities of users, or workgroup heads in librarian organizations).
3. *Social-Informational* dimension, which concerns the diverse conceptual models needed to support the work of such different community of users, and their impact on the documents (e.g. a virtual museum curator has to describe the items of a document taking into account constraints imposed by user interfaces in order to show tool-tips effectively when a visitor moves the mouse over a particular exhibit in the scene). New roles can have a different view of documents, so the digital library should provide them the information required with the format more suitable to their needs. This is true also for future user roles not discovered during the development stage of the library.

[3]In the original Yates' model, these domains were called *documents*, *technology*, and *work*, respectively.

4 EVOLUTION PROBLEMS

Now we discuss the three classes of open issues identified in previous Section 3, introducing some representative examples.

4.1 The data evolution problem

The first prototype of E-Dvara provides users a flexible way to define and update the metadata associated to each project representing a digital archive. In particular, users can define a set of *schemata* which supplies the structure adopted for storing documents. Each schema is expressed in terms of *fields*, *data types* and *constraints*. Metadata definition can take place every time during the digital collection lifecycle, leading to the problem of correctly handle the evolution of data. In fact, such an iterative schema definition process is based on a continuous refinement of activities, executed (by schema experts) to increase its effectiveness in data preservation and representation.

Such an approach, however, requires the introduction of methodologies devoted to perform data validation accordingly to the evolution of the schema used to represent the specific content the users want to store. In fact, each schema update should be properly spread to the previously validated archives, in order to automatically adapt the existing content to the new schema (or, if this is not possible, to provide modelers the feedback necessary to manually fix the problem).

Examples of this process can be defined considering the data model illustrated in Figure 2. A more exhaustive discussion of this data representation model will be provided in Section 5.1; in this moment, we only suppose that, starting from the

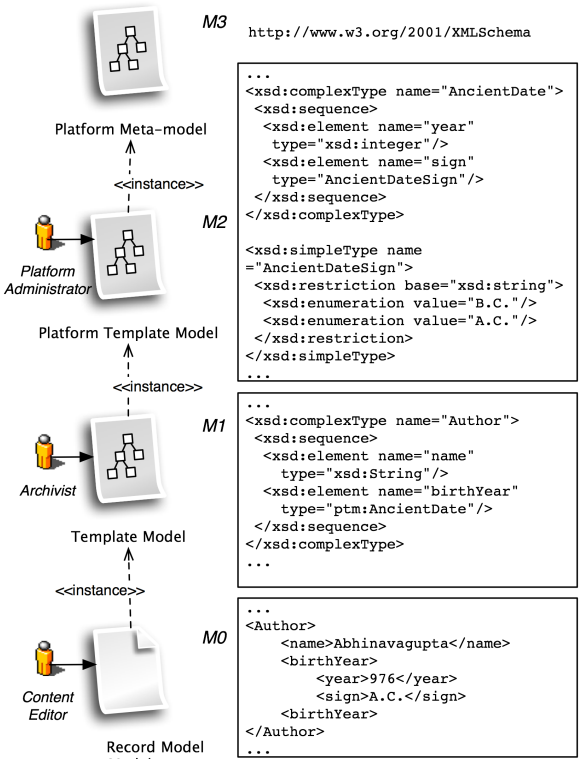

Figure 2: XML Representation of Data Model.

schema at level M1, we want to insert more information into the Author element by *adding* new fields (`Place of Birth`, `Nationality`), *modifying* existing fields (splitting `Name` into `Suffix`, `First Name`, `Last Name`, `Prefix`) or *removing* fields which are considered no more useful. Moreover, we may also be interested in moving from the `AncientDate` format to a type representing dates in a modern way. Clearly, in order to handle these issues, a set of dedicated tools and methodologies should be provided. The evaluation we performed in the E-Dvara project has suggested us that the flexibility of *mutable templates* (i.e. evolving schemata) must be considered as an essential feature for our platform. At the same time, preservation and adaptation mechanisms are also required, especially when evolution concerns large and mature collections of existing contents.

4.2 Technical infrastructure adaptation

One of the recurring issues we have faced using the first prototype of E-Dvara was the request for integrating new heterogeneous functional modules at the top of the digital library (e.g. virtual museums, metasearch engines, or applications for mobile devices). These requests posed unforeseen challenges on the software infrastructure. For each new application, we needed to rewrite a lot of ad-hoc business logic, without mention the fact that we had to duplicate some services only to adapt them to a new required programming interface. Another tricky aspect was sharing similar behaviors and data located in different components which were based on incompatible communication protocols. These issues clearly demanded for a *reusable integration layer* that we lacked as part of our software architecture. Moreover, we have learned also that integrating several applications in a common environment requires a substantial investment in understanding and implementing their *orchestration*, in order to handle incompatibilities between different business logics in a standard and transparent way. Orchestration enables functional chunks (exposed as Web services) to be strung together in predefined patterns which map to business processes and workflows. In conclusion, we failed to recognize the importance of these patterns, which are the building blocks to describe the interaction between different applications by identifying messages, branching logic and invocation sequences. If we had realized that since the beginning, composing together heterogeneous functionalities would be simpler to achieve. Moreover, also the deployment would be expedited, especially

by moving that orchestration logic from the inside of a component to an external configuration file (i.e. by means of a XML description file associated to each component), enabling a flexible and dynamic setup.

4.3 Coordination issues of different user roles

The integration of different applications on the top of the existing software infrastructure, especially when related to diverse domains, were typically the manifestation of new requirements involving user roles and the policies they were subjected to during information access. One example of this situation is an external service which, based on its own data management policy, states *when* a particular workflow is required to organize the archived contents. In E-Dvara, a *workflow* expresses a set of roles, related activities and constraints that define together the structure of the information management process. Furthermore, several tasks in the Social domain may require to access, organize and enrich existing contents designed and generated by different users. As an example of such a workflow, consider the curator of a digital museum which has to arrange a new gallery, composed by paintings, ancient books and movies hosted in three projects and owned by three different users. Consider now the case in which the curator wants to incorporate in the same gallery a set of features to search, organize and enrich the existing records, by adding new fields describing the position each item will have in the 3D rendering of the virtual museum. Moreover, final users may also be interested in improving the quality of the exhibition, by creating new relations between the existing content (e.g. opinions and links to a specific content in a typical Web 2.0 style).

These scenarios pose several issues that must be faced in order to provide flexibility in the way data management is achieved. Such issues concern *intellectual property* (Is the user allowed to use a specific content?), *strong coupling between projects* (How do the schemata of project B evolve according to the evolution of both schemata and data in project A, if any dependency between them exists?) and *strong coupling between workflows* (Do the activities in the workflow A overlap those in workflow B?).

5 HANDLING EVOLUTION

This section proposes a distributed approach to handle the evolution problems discussed in previous Section 4.

5.1 Evolution along the Informational - Technological dimension

In order to handle the evolution problems concerning the changes in data format and schemata described in Section 4.1, we propose here a four-layer data representation model (Figure 2). At the bottom of the hierarchy, we place the *records* (level M0, Record Model), aimed at representing the archived data (documents). A record is a particular instance of a document stored in the digital platform. Every document must also conform to a *document template* (level M1, Template Model), which provides structural definitions (e.g. the document contains the `Title`, `Author`, and `Date` fields) and constraints (e.g. the `Data` field must conform to the `mm/dd/yy` format or the `Title` field is mandatory). Document templates are themselves conformed to *platform template* (level M2, Platform Template Model) devoted to define both business rules and data types the archivists can use to build document templates (e.g. each record in every archive must contain the `Creation Date` and `Owner` fields). Finally, platform templates are instances of a more general layer, the *platform metamodel* (level M3, Platform Meta-Model), which defines a set of common low-level structures (e.g. primitive data types as `xsd:String`) and operations (e.g. data sequencing) available in order to define more complex data structures. This level is that of the OMG XML Schema specifications.

The overall data model involves the interaction with three different actors:

- *Content editor*, devoted to data entry, with respect to a specific document template; however, he is not allowed to perform any template change.
- *Archivist*, devoted to document templates definition.
- *Platform administrator*, devoted to the management of platform templates (e.g. the templates provided by archivists should be validated against the platform template model each time they are created/modified or when the platform template model itself is updated).

This hierarchical data model provides *automatic data validation policies* which play a central role in our vision. Indeed, validation is applied both to the templates and (recursively) to all the records stored in the platform archives. Templates which do not respect the business rules defined in the platform template model should be manually updated by either archivists or content providers in order to become consistent. This type of validation is propagated then to the platform

meta-model (level M3) which acts as a template for the platform template model (level M2).

In order to develop the proposed model, we present an implementation approach based on the XML technology and standards[4], focusing our attention on the features provided by XML Schema.

5.2 Evolution along the Technological - Social dimension

In order to handle the evolution issues concerning the adaptation to new requirements such as the integration of heterogeneous services described in Section 4.2 we base the second prototype of E-Dvara according to a Service-Oriented Architecture (SOA) model (Figure 3), characterized by:

- The introduction of an explicit *Integration layer*, which forms the "architectural glue" that brings the digital library beyond the scope of a single application, unifying the interfaces of different subsystems into the same interoperable environment.
- The migration toward *services*, which are more autonomous, composable, and stateless than traditional software components (e.g. dynamic libraries).
- The adoption of a *peer-to-peer, message-based communication protocol* supported by the *Enterprise Service Bus* (ESB), which connects disparate applications orchestrating their interactions, mediates their incompatibilities, and makes them broadly available as services for additional uses.

The standard set of Web service technologies (XML, SOAP, WSDL) provides the means to describe, locate and invoke a Web Service as an entity in its own right. However, it is often necessary to compose different services with some logic in order to complete a task, as described in Section 4.2. This is where orchestration plays a crucial role, deployment sophisticated and complex Web services as a whole unit of functionality. Hence, the orchestration engine (the ESB component) acts as a centralized authority to coordinate interaction between services and applications.

At the top of our SOA architecture we have placed applications such as administration interfaces to manage users and archives, publication interfaces to produce new content in the digital library, or virtual museums to exhibit a document archive in a "museum-like" setting. All these heterogeneous modules can exploit any reusable service available in the Integration layer, (e.g. to perform searches in the platform archives). Services are orchestrated by the ESB which simplifies both integration and reuse of business components within the SOA system. Finally, at the bottom of the architecture are placed the archives, which are managed by two dedicated applications: the *Archive Manager* which stores and retrieves documents, and the *Policy Manager* which manages users, accessing policies, and projects (which organize the archives). The Archive Manager isolates the business logic needed to realize the data-model described in Section 5.1, whereas the Policy Manager implements the data validation rules, decoupling them from other architectural components. This is an example of *separation of concerns* which goes far beyond the decoupling of user interface from data representation, and characterizes the whole E-Dvara architecture. The overall architecture, organized according to the SOA model, is more suitable to respond to the integration requests we have to face in E-Dvara.

5.3 Evolution along the Social - Informational dimension

The introduction of mutable templates in content representation provides the ability to update a schema during the whole life-cycle of a data collection, but leads also to several challenges, the most important of which is the evolution and re-evaluation of existing archives. In this section, we introduce a *multi-agent approach* to tackle the problem, aimed (when possible) to automatically resolve evolution issues.

The levels from M1 to M3, proposed in Figure 2, can be affected by updates during the digital library life-cycle. In particular, such updates can involve XML Schema definitions (level M3, with a low frequency), Platform Template Models (level M2, with a low-medium frequency) and Template Models (level M1, with a rather high frequency).
Each schema is connected by a dependency bond with the schemata on its top for validation purposes. However, in a collection one level can be related to another also by means of relations between different data types (e.g. an instance of the template `Book` in M1 can be related with one or more instances of the template `Author` in M0). At the same time, we can also have a relation connecting templates in different collections (e.g. instances of the template `GalleryRoom` in a virtual museum application can be related with instances of `Book` and `Painting` templates taken from different collections). Hence, such dependencies requires evolution mechanisms that must be propagated both in a specific level and across multiple levels.

[4]See for more details: http://www.w3.org/XML/Schema

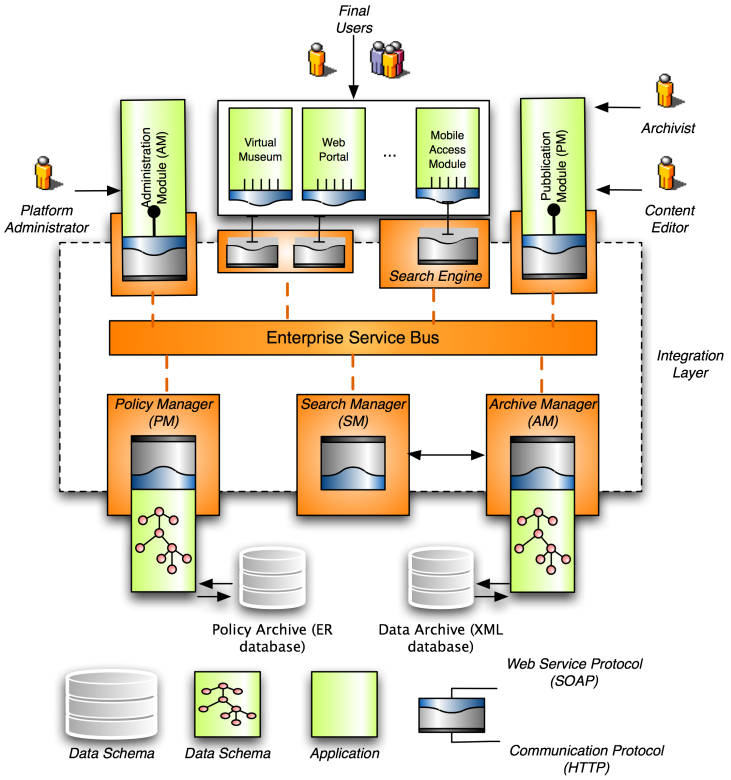

Figure 3: The architecture of the E-Dvara platform

This propagation mechanism is achieved by means of a multi-agent system. Each agent is assigned to a specific schema, monitoring its evolution; an agent can interact with other agents assigned to depending schemata, send them messages and apply evolution to the instances of its schema.

A *coordinator agent* is assigned to each instance of the platform, in order to monitor the updates of the Platform Template Model and to activate the agents connected to each schema when required. The coordinator agent is also devoted to the creation of a new agent every time a new schema is defined (even if it does does not act directly on data because such task is delegated to agents located at level M1).

A *schema agent* is devoted to the evolution of contents related to a specific template at level M1. They can perform a set of actions on the existing data, accordingly to the updates affecting related schemata.

Agents perform several evolutionary operation on data, in order to preserve data validity and, at the same time, to prevent archivists and content editors to spend a lot of time re-entering the whole set of existing contents. In (Guerrini et al., 2005; Guerrini et al., 2007) a complete taxonomy of updates, which can affect a generic XML schema, is described; actually only a subset of the listed operations has been implemented in E-Dvara, covering the set of updates which can be performed by archivists. For example, we provide the commodities to rename or adding elements and attributes of the Template Model (level M1).

In order to cope with the complexity of the evolution tasks and the amount of data yet available in E-Dvara, our attention is focused on simple updates which commonly occur during the life-cycle of a collection. For example, a typical evolution task is represented by the extraction of a vocabulary (a closed list of predefined strings) from the set of values assigned to a free-text `String` element. In our experience such an update is rather frequent, specifically when we are not able to know a priori *all* the values assignable to a specific element. In this case, when an archivist decides to change the type of the element `Name` from `String` to `Vocabulary`, the agent assigned to that schema should access each instance of the template and perform a `change_item_type`, verifying if the old values assigned to `Name` are validated with respect to the values admitted by the new element type. When this task is completed, the agent should notify the schema updates to the related agents (according to the dependency chain between schemata), in order to

grant the consistency of any inter-dependent data.

6 CONCLUSIONS

In this paper we have extended an existing conceptual model for digital libraries, introducing the notion of evolution dimension and describing our proposal along three dimensions: Informational - Technological, Technological - Social, and Social - Informational. This characterization comes from the lessons learned during the experimentation with our E-Dvara platform.

Now we are working to complete the second prototype which embodies the improvements described in the paper.

Our future plans include a validation of the overall prototype in different areas, concerning the exploitation of both information and services by means of mobile applications, virtual museums, and Web 2.0 environments.

REFERENCES

Bainbridge, D., Buchanan, G., Mcpherson, J., Jones, S., Mahoui, A., and Witten, I. (2001). Greenstone: A platform for distributed digital library applications. In *ECDL '01: European Digital Library Conference*, pages 137–148. Springer.

Barkstrom, B., Finch, M., Ferebee, M., and Mackey, C. (2002). Adapting digital libraries to continual evolution. In *JCDL '02: Proceedings of the 2nd ACM/IEEE-CS Joint Conference on Digital Libraries*, pages 242–243. ACM.

Bawden, D. and Rowlands, I. (1999). *Understanding digital libraries: towards a conceptual framework*. British Library Research and Innovation Centre.

Bekaert, J., Liu, X., and Van de Sompel, H. (2005). adore: A modular and standards-based digital object repository at the los alamos national laboratory. In *JCDL '05: Joint Conference on Digital Library*, pages 367–367. ACM.

Guerrini, G., Mesiti, M., and Rossi, R. (2005). Impact of xml schema evolution on valid documents. In *WIDM '05: Proceedings of the 7th annual ACM International Workshop on Web Information and Data Management*, pages 39–44. ACM.

Guerrini, G., Mesiti, M., and Sorrenti, M. A. (2007). Xml schema evolution: Incremental validation and efficient document adaptation. In *Database and XML Technologies, 5th International XML Database Symposium*, pages 92–106.

Lagoze, C., Payette, S., Shin, E., and Wilper, C. (2005). Fedora: An architecture for complex objects and their relationships.

Lutzenkirchen, F. (2002). Mycore - ein open-source-system zum aufbau digitaler bibliotheken. *Datenbank-Spektrum*, 4:23–27.

Rowlands, I. and Bawden, D. (1999). Digital liraries: A conceptual framework. *Libri*, 49:192–202.

Tansley, R., Bass, M., Stuve, D., Branschofsky, M., Chudnov, D., McClellan, G., and Smith, M. (2003). The dspace institutional digital repository system: current functionality. In *JCDL '03: Joint Conference on Digital Libraries*, pages 87–97. IEEE.

Witten, I., McNab, R., Boddie, S., and Bainbridge, D. (2000). Greenstone: A comprehensive open-source digital library software system. In *ICDL '00: International Conference on Digital Libraries*. ACM.

Yates, J. (1989). *Control through communication*. The Johns Hopkins University Press.