A Cooperative-Relational Approach to Digital Libraries

A. Malizia[1], P. Bottoni[2] S. Levialdi[2], and F. Astorga-Paliza[1]

[1] Computer Science Department, Universidad Carlos III de Madrid.
Avda. de la Universidad, 30. 28911 Leganés, Madrid (Spain)
{amalizia,jastorga}@inf.uc3m.es
[2] Computer Science Dept., Universitá "Sapienza" of Rome,
Via Salaria 113, 00198 Rome, Italy
{bottoni,levialdi}@di.uniroma1.it

Abstract. This paper presents a novel approach to model-driven development of Digital Library (DL) systems. The overall idea is to allow Digital Library systems designers (e.g. information architects, librarians, domain experts) to easily design such systems by using a visual language. We designed a Domain Specific Visual Language for such a purpose and developed a framework supporting it; this framework helps designers by automatically generating code for the defined Digital Library system, so that they do not have to get involved into technical issues concerning its deployment. In our approach, both Human-Computer Interaction and Computer Supported Collaborative Work techniques are exploited when generating interfaces and services for the specific Digital library domains.

1 Introduction

Digital Libraries are complex information systems involving many different areas: Library and Information Science (LIS), Information Retrieval (IR) and Human-Computer Interaction (HCI), to name a few. Google books, the ACM Portal, or Springer on line are examples of Digital Libraries that we use on a dayly basis. But from the designer point of view, there is a need for case tools or modeling support for describing not only the contents but also the interactions and the collaboration work that can happen within such complex systems. For example, scenario or activity-based approaches can be mutuated from HCI in order to model the society of users cooperating within a Digital Library. For example, a scenario can happen in which users have to concurrently access the same document to contribute to its tagging, or to provide advanced services through shared content. Indeed, services like: cross-references, focus groups on special subjects, deployment of collective tagging can be of great interest to Digital Libraries users. Moreover, there are mainly two categories of designers involved in such systems: Librarians and Information Scientists, plus Software engineers (experts in various fields from Information Retrieval to Database Management Systems). These categories of users are generally in contrast when deploying Digital Libraries. Librarians are the domain experts able to deal with faceted categories of documents, taxonomies and document classification, while engineers usually concentrate on services and code development. With our framework, we aim to propose an approach that is suitable for both, allowing librarians to categorize and model the documents as well as their collections, and software engineers to focus on service development and requirements.

The paper is organized as follows: Section 2 presents an overview of the definitions of Digital Library systems and previous work relevant to the presented approach. Section 3 is about modelling Digital Libraries environments considering a model-driven approach for collaborative work scenarios. Section 4 explains the elements of the meta-model at the core of our work. Section 5 illustrates a working example of a Digital Library automatically generated by our framework, and describes more general applications, while Section 6 draws the conclusion and discusses future works.

2 Background and Related Work

There are many definitions of DLs, for example the Delphi study by Kochtanek *et. al.* [1] of digital libraries coalesced a broad definition: organized collection of resources, mechanisms for browsing and searching, distributed networked environments, and sets of services objectified to meet users' needs. The President's Information Technology Advisory Committee (PITAC) Panel on Digital Libraries treats digital libraries as the networked collections of digital text, documents, images, sounds, scientific data, and software, that make up the core of today's Internet and tomorrow's universally accessible digital repositories of all human knowledge. Underlying all these definitions there is a consensus that digital libraries are fundamentally complex. Such complexity is due to the inherently interdisciplinary nature of this kind of systems. Digital libraries integrate findings from disciplines such as hypertext, information retrieval, multimedia services, database management, and human-computer interaction [2]. Designers of digital libraries are most often library technical staff, with little to no formal training in software engineering, or computer scientists with little background in the research findings of information retrieval or hypertext. Thus, digital library systems are usually built from scratch using specialized architectures that do not benefit from digital library and software design experiences. Wasted effort and poor inter-operability can therefore ensue, raising the costs of digital libraries and risking the fluidity of information assets in the future. Formal models and theories are crucial to specify and clearly, understand the characteristics, structure, and behavior of complex information systems. It is not surprising that most of the disciplines related to digital libraries have underlying formal models that have properly steered them: databases [3], information retrieval [4, 5], and hypertext and multimedia [6]. Furthermore, formal models for information systems can be used for the design of a real system, providing a precise specification of requirements against which the implementation can be compared for correctness. Currently, there is a huge bibliography on digital libraries, while there are only a few papers dealing with DL within CSCW environments.

The Digital Libraries Group at Universidad de las Americas-Puebla (UDLA - Mexico) [7] introduced the concept of personal and group spaces which are relevant in a CSCW domain in the DL system context. Users can share information stored in their personal spaces or share their agents for allowing other users to perform the same search on the document collections in the DL.

In [8], the authors describe a formal foundation theory, on digital libraries, called 5Ses based on the following concepts: streams, data structures, spaces (for the resource space), scenarios, societies. This approach is an evidence of a good modeling endeavour but it doesn't specify formally how to derive a system implementation from the model. In the CRADLE framework we chose the E/R formalism, mainly for two reasons: it is powerful and general enough for describing digital libraries' models (at least it is frequently used for modeling DBMS applications which are foundations for digital libraries) [9], and is supported by many tools as a meta-modeling language. Although most approaches to entity-relationship modeling do not deal deeply with dynamic aspects, because the entity-relationship approach is used for modeling static structure though it ought not to be separated from the behavior alone. Temporal entity-relationship extensions [10] add dynamic aspects to the entity-relationship approach, but most of them are not directed to object-oriented approaches. Recently the advent of object-oriented based technology calls for and demands information systems design approaches and tools resulting in object-oriented systems. These considerations drove the research towards modeling approaches like the Unified Modeling Language (UML) [11]. Since UML metamodel is specified by a combination of UML class diagrams (abstract syntax), OCL (well-formedness rules) and English (detailed semantics), it lacks the rigor of a language precisely defined using formal language techniques. The imprecision of the UML specification has undesirable consequences for users, since engineers might use implementation decisions that are inconsistent with the specification and other implementations.

3 Modeling Collaborative Scenarios in Digital Libraries

Our approach generates code from tools built after modeling a digital library (according to the rules defined by the proposed meta-model); we use an automatic transformation and mapping from model to code so as to generate software tools for a given digital library model. We call our methodology **Cooperative-Relational Approach to Digital Library Environments** (CRADLE model). In CRADLE, the specification of a digital library encompasses four complementary dimensions: 1. multimedia information supported by the DL (Collection Model); 2. how that information is structured and organized (Structural Model); 3. the behavior of the DL (Service Model); and the different societies of actors; 4. groups of services that act together to carry out the DL behavior (Societal Model). Initially, a DL designer is responsible for formalizing a conceptual description of the digital library using the meta-model concepts. This phase is normally preceded by an analysis of the DL requirements and characteristics

(Figure 1a). Model specifications in CRADLE are then fed into a DL generator (written in Python for ATOM3 [13]), to produce the tailored DL, suitable for specific platforms and requirements (Figure 1b). We chose ATOM3 for its meta-meta-model specification which describes the basic elements that can be used to design a meta-model modelling formalism. If newer concepts and structures need to be introduced, they can be modelled at a meta-meta-level. The advantage of the AToM3 approach is the flexibility that can be achieved. In fact, by adapting the model of a modelling formalism, and automatically generating a prototype of the modelling environment, design choices can be rapidly evaluated.

These are built upon a collection of stock parts and configurable components that provide the infrastructure for the new DL (Figure 1c). This infrastructure includes the classes of objects and relationships that make up the DL, and processing tools to create/load the actual library collection from raw documents, as well as services for searching, browsing, and collection maintenance. Finally, the `LibGen` module (Figure 1d) generates tailored DL services code stubs by composing and specializing components from the component pool. CRADLE is in its alpha version but we have already used it to build pilot systems and prototypes.

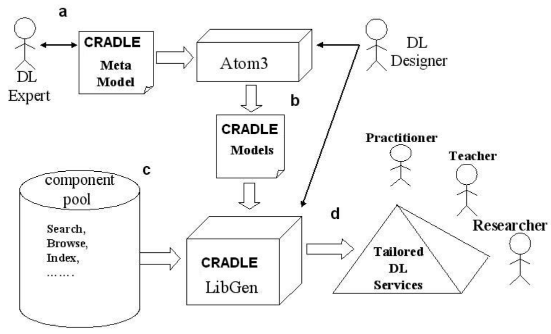


Fig. 1. The CRADLE scheme

Most of the CRADLE model primitives are defined as XML based elements, which can enclose other sublanguages that help define DL concepts. The XML User Interface Language (XUL) is used to represent appearance and visual interfaces [14], while the XDocLet [15] (also XML based) standard is used for deploying service templates.

4 The Cradle Meta-model

In the CRADLE formalism, the specification of a DL, includes: multimedia documents supported by the DL - Collection Model; how that information is structured and organized - Structural Model; the behavior of the DL - Service Model; and the different societies of actors and groups of services that act together to carry out the DL behavior - Societal Model. In our approach a society is an instance of the CRADLE model defined according to a specific collaboration framework in the digital library domain. A society is the highest-level component of a digital library, which exists to serve the information needs of its set of actors and to describe the context it is used in. Digital libraries are used for collecting, preserving, and sharing information artefacts between society members. In fact, cognitive models for information retrieval [16, 17], for example, focus on users information-seeking behavior (i.e., formation, nature, and properties of a users' information need) and on the ways in which information retrieval systems are used in operational environments. After carefully reviewing literature on digital libraries topics, we selected basic entities among the facets from the categorizations presented in [8]. In fact, in the digital library context, we can model actors as the users of digital libraries. Actors interact with the DL through services (interfaces) that are (or can be) affected by the actors preferences and messages (raised events). Another class selected from the proposed study are Activities. Activities within cooperation digital libraries consist of: collecting, creating, disseminating, evaluating, organizing, personalizing, preserving, requesting, and selecting. All these activities can be described and implemented using scenarios and appear in the DL setting as a result of actors using services (thus societies). Digital libraries can contain repositories of documents (Components), information, data, metadata, relationships, logs, annotations, and user profiles, all of which can be interpreted as distinct types of digital objects, according to their specific structure, metadata, and relation. The Socio-economic class represents what surrounds the DL. This facet is mainly related to the societal aspects of the DL and their interactions abstracting aspects surrounding the DL such as: policies, economic issues, standards, and organizational attributes. Finally the Environment class is intended as the contexts DLs are embedded in. The environment involves a set of spaces (e.g., the physical space, or a concept space defined by the words of a natural language) that defines the use and the context of a DL.

In a previous paper [18] we presented an early model called SADDLE, which had limitations with respect to the complexity of the Digital Libraries which could be developed with it, due to the incomplete specification of some elements. In the new meta-model presented here, we introduced for instance the notion of Document detached (but related) to the Collection and introduced a mechanism for managing the responses to events that describe in the following paragraph. In the CRADLE model a Society is a group space made of several personal spaces, and, agents are modeled as services interacting with actors and collections (resources). The idea of personal and group spaces and agents is interesting and stimulating; nevertheless a concurrent and cooperative task model should be included in digital libraries both for managing services (synchronous/asynchronous) interactions and for specifying operative scenarios as we describe in the following paragraph. This means that the CRADLE approach aims at filling the gap to support models for the design of user interface

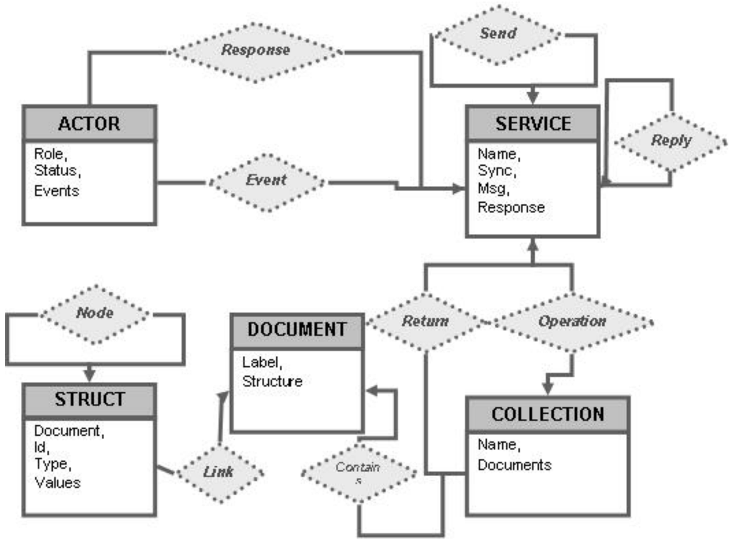


Fig. 2. The CRADLE meta-model with the E/R formalism.

and interaction among collaboration in digital libraries systems. The meta-model in Figure 2 includes the basic entities:

- The Actor entity has three attributes: Role, Status and Events.
  - Role: a description of the role (i.e. librarian, server,)
  - Status: None (N.A.), Active (present in the model and actively generating events), Inactive (present in the model but not generating events,Sleeping (present in the model and awaiting for a response to a raised event)
  - Events: describes a list of events that can be raised by the Actor, or received as response message from a Service (which is treated as an event by an Actor). Examples of events (for a library environment) can be: borrow, reserve, return, etc.
- The Service entity has four attributes: Name, Sync, Events, Responses.
  - The Name attribute is a string representing a textual description of the service.
  - The Sync attribute states whether the service employs a synchronous or asynchronous communication, and has two possible values: wait (synchronous) or nowait (asynchronous).
  - The Messages attribute is a list of messages that can trigger actions among services (tasks); for example valid or not valid in case of a parsing service.
  - The Responses attribute contains a list of response messages that can reply to raised events; they are used as communication mechanism among actors and services.

- The attributes of Collection are: Name and Documents; Name is a string which specifies the logical name of the Collection, while Documents is a list of couples made of Document name and Document Label (a pointer to the Document entity).
- In the Document entity two different attributes are present: Label and Structure.
  - The Label defines a textual string which can be referenced by a Collection entity. We can view it as a document identifier, specifying a class or a type of documents.
  - The Structure defines its semantics and area of application. For example, any textual representation can be seen as a set of characters, so that text documents, such as scientific articles and books, can be considered as a structured set of elements.
- Structures are represented as graphs and the Struct entity (a vertex) contains four attributes: Document, Id, Type, Values.
  - The Document attribute is a pointer to the Document entity the structure refers to.
  - The Id is a unique identifier for structures elements.
  - The Type attribute takes three possible values: Metadata, Layout, and Item. Metadata indicates that the structural element is a content descriptor, for instance title, author, etc. Layout indicates that the structural element is mapped on a layout, such as: left frame, columns, header, etc. Item indicates a generic structure element that can be used for extending the model while keeping it general.
  - The Values attribute is a list of single or multiple values the structure element can take; it describes the element content, like for instance title, author, etc.

The Relationships between the entities shown in Figure 2 are quite self-explanatory. Actors interact with Services by an event-driven communication model. Services are connected to each other by synchronous or asynchronous messages (send and reply). Services can perform operations (like: get, add and del) on Collections and these operations return Collections of Documents as results. Documents are contained in Collections and Struct elements are connected to each other as nodes of a graph representing metadata structures associated to documents. In the next section we present a basic example of how to use our framework to generate a simple Digital Library.

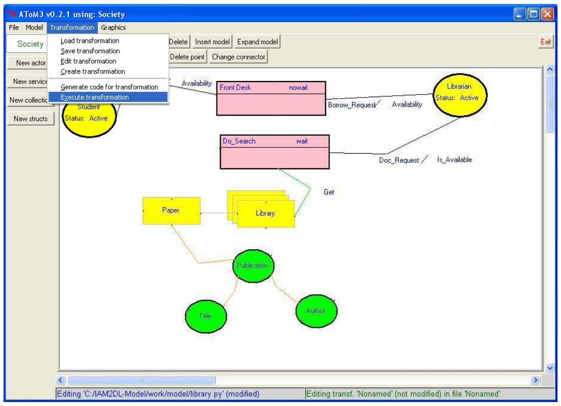
5 Generating Digital Library Environments

As a first step in designing the DL environment in the CRADLE framework, designers model the *Society* involved in the specific scenario. We define a running example, called *Library*, modeling a simple digital library environment to show the overall process, starting from the basic entities of the model. The *Actors* (represented by circles) involved in this *Library* are: *Students* and *Librarians.*

The digital library *Collection* (represented by multiple rectangles) consists of *Digital Paper Documents* structured with *Publication, Author* and *Title* metadata information (*Struct entities*).

There are two basic services available in this example, the *Front Desk* and the *Search* services. The *Front Desk* is responsible for managing communication between *Students* and *Librarians*, while the *Search* service executes queries on the digital library. In Figure 3, the CRADLE environment is shown together with the defined entities. The rectangles render the *Services* appearance, while the single rectangle connected to a *Collection* represents a *Document* entity; the circles linked to the *Document* entity are the *Struct* (metadata) entities.

The represented scenario is about a *Student* trying to borrow a *Paper* from the *Library*; she interacts with the *Front Desk* service requesting a paper and obtaining a response message about its availability within the digital library. The *Front Desk* service is asynchronous (see Section 4) and forwards the borrow request (Borrow_Request) to the *Librarian* actor. The *Librarian* sends a Doc_Request message to the *Search* service (Do_Search). The *Search* service is synchronous (see Section 4). It queries the document collection looking for the requested document and waits for the *get* result (a collection of documents) to send the response back. The service returns an *Is_Available* boolean message which is then propagated as a response to the *Librarian* and eventually to the *Student* (see Figure 3).



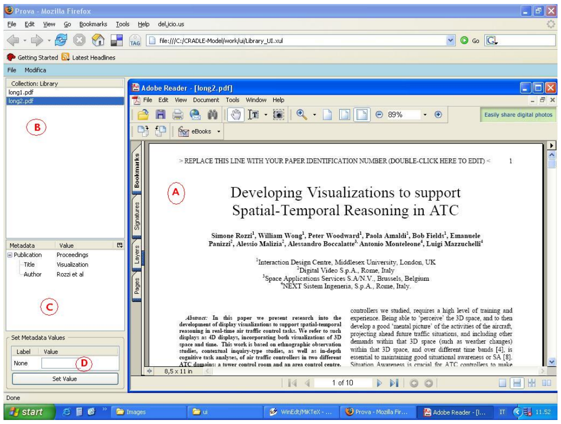
Editing 'C:/IAM2DL-Model/work/model/library.py' (modified)

Editing transf. 'Nonamed' (not modified) in file 'Nonamed'

Fig. 3. Starting the code generation process by transformation execution.

When the library designer has built the model, if the *execute transformation* menu item is selected, the framework runs the transformation process, executing the code generation actions associated with the entities and services represented in the model.

The user interface generation occurs according to the XUL and XDocLet templates and the entities defined in the model. In this example the generated UI is presented in Figure 4.



file:///C:/CRADLE-Model/work/ui/Library_UI.xul

Collection: Library

long1.pdf

long2.pdf

Metadata	Value
Publication	Proceedings
Title	Visualization
Author	Rozzi et al

Set Metadata Values

Label	Value
None	

Set Value

> REPLACE THIS LINE WITH YOUR PAPER IDENTIFICATION NUMBER (DOUBLE-CLICK HERE TO EDIT) < 1

Developing Visualizations to support Spatial-Temporal Reasoning in ATC

Simone Rozzi[1], William Wong[1], Peter Woodward[1], Paola Amaldi[1], Bob Fields[1], Emanuele Panizzi[2], Alessio Malizia[2], Alessandro Boccalatte[3], Antonio Monteleone[4], Luigi Mazzuchelli[4]

[1]Interaction Design Centre, Middlesex University, London, UK
[2]Digital Video S.p.A., Rome, Italy
[3]Space Applications Services S.A/N.V., Brussels, Belgium
[4]NEXT Sistem Ingeneria, S.p.A., Rome, Italy.

Abstract: In this paper we present research into the development of display visualizations to support spatial-temporal reasoning in real-time air traffic control tasks. We refer to such displays as 4D displays, incorporating both visualizations of 3D space and time. This work is based on ethnographic observation studies, contextual inquiry-type studies, as well as in-depth cognitive task analyses, of air traffic controllers in two different ATC domains: a tower control room and an area control centre.

controllers we studied, requires a high level of training and experience. Being able to 'perceive' the 3D space, and to then develop a good 'mental picture' of the activities of the aircraft, projecting ahead future traffic situations, and including other demands within that 3D space (such as weather changes) within that 3D space, and over different time bands [4], is essential to maintaining good situational awareness or SA [8]. Situation Awareness is crucial for ATC controllers to make

Fig. 4. The UI generated by CRADLE transforming the Library model in XUL code.

The generated UI is based on the template code, enriched with information from the modeled entities. On the right side of Figure 4(A) the document area is presented according to the XUL template. Documents are managed in this area according to their MIME type; in this example, since the documents are *PDF* files, they are loaded with the appropriated *Acrobat Reader* plug-in. On the left column of the UI are three basic boxes. The *Collection* box, Figure 4(B), presents the list of documents contained in the *Collection* specified by the *Documents* attribute of the *Library Collection* entity, and allows users to interact with documents. After selecting a document by clicking on the list, it is presented in the document area (Figure 4(A)) where it can be managed. In the *MetaData* box, Figure 4(C), the tree structure of the metadata is depicted according to the metadata categorization modeled by the designer. The XUL template contains all the basic layout and action features for managing a tree structure. The generated box contains the parent and child nodes according to the attributes specified in the corresponding *Struct* elements included in the model. The user can click on the root for compacting or exploding all the tree nodes, and by selecting one, the UI activates the *MetaData operations* box, Figure 4(D).

After the selected metadata item is presented in the "set MetaData Values" box, the user can edit its values and by clicking on the "set value" button save this information. The "set value" button operation, not only saves the metadata information, but also displays it in the intermediate box (tree-like structure) for changing the UI visualization according to the new values.

The code generation process for the designed services is based on the XDoclet templates. The CRADLE framework generates the code for the messages and events as designed in the *Library* model.

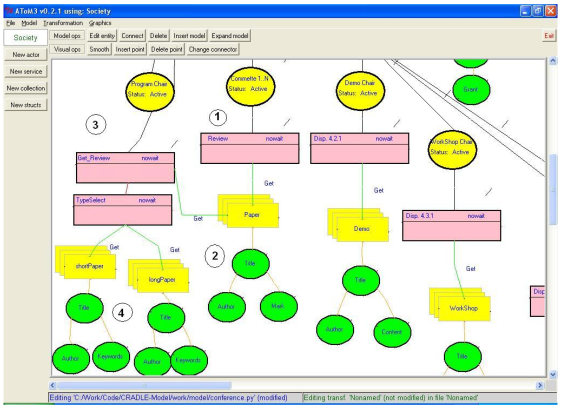
In particular, for the *Front Desk* service, a *message listener* template is used to generate the Java code. The *Actors* classes are also generated by using the services templates since they have attributes, events and messages just like the services. The *Do_Search* service code is based on the producer and consumer templates since it is synchronous by definition in the modeled scenario.

The overall code generation process is based on template code snippets generated from AToM environment graph transformation engine, following the generative rule specified by our meta-model.

To generate the tree structure for the metadata, the Transformation action is invoked and some Python code is executed loading XUL section templates from external files; exploring the metadata tree in the visual model with a Breadth-first search and attaches the XUL code for displaying the *Struct* node in the appropiate position within the graph structure in the visual interface according to the metadata String specified in the model.

Summarizing, in the CRADLE approach the code generation process is based on templates provided for basic interfaces, services and actors behaviors.

We have explored a variety of personal and group interfaces and environments to place the digital collections' wealth at the user's disposal. For instance, we considered the case of an international scientific conference management with all the roles played by the chairs and committees in reviewing and accepting (or rejecting) different kinds of documents. Figure 5 shows en extract of a detailed conference model view in the CRADLE framework.



Editing 'C:/Work/Code/CRADLE-Model/work/model/conference.py' (modified)

Editing transf. 'Nonamed' (not modified) in file 'Nonamed'

Fig. 5. Extract from the CRADLE visual model for the reviewing process in an academic conference environment.

As shown in Figure 5, the Committee actor (this is an example of a generic Committee that can change depending on the task purpose), by using the Review service, uploads the reviewed papers (Figure 5(1)) in the Paper Collection, populating the corresponding metadata (Figure 5(2)): Title, Author, Mark (representing the review synthesis, e.g. fair, poor, good, etc.). Among the Program Committee actor duties is the management of the submitted reviews (get other Committees reviews) and making the final decision for acceptance and form of publication (short or full papers). The Program Committee actor gets the submitted reviews using the Get Review service and selects type of publication, such as short or full (long) by invoking the TypeSelect service (Figure 5(3)). Finally, documents are uploaded into the different collections with their basic metadata (Figure 5(4)): Title, Author, Keywords. Some components are ready to be incorporated into regular user activities (searching, browsing, tagging), others are at different stages of development, from preliminary prototypes to usability testing.

6 Conclusion and Future Work

Summarising, Digital Libraries (DLs) are extremely complex information systems that integrate findings from disciplines such as hypertext, information retrieval, multimedia services, database management, and human-computer interaction. Designers of DLs are often multidisciplinary teams, which include library technical staff and computer scientists. Wasted effort and poor inter-operability can therefore ensue (raising the costs of DLs and hindering the fluidity of information assets). Examining the related bibliography we noted that there is a lack of tools or computer-aided systems for designing and developing Cooperative DL systems. Moreover, there is a need for modeling interactions among DL systems and users (as proposed in the HCI field) such as: scenario or activity-based approaches.

The CRADLE framework aims to fill this gap by providing a meta-model based approach for generating visual interaction oriented tools for DLs. We experimented with it within a group of graduate students from the School of Library and Information Science (Scuola Speciale per Archivisti e Bibliotecari - SSAB), at University La Sapienza of Rome, Italy. They future work will be as librarians and information architects and thus their help was crucial in developing our approach. Morover we involved some graduate students from the Computer Science Dept. at University La Sapienza of Rome, Italy, who worked as the service engineers. The early results (with documents in Italian) were very encouraging but further investigation is needed.

In fact, recently, AToM3 has been provided with the possibility to describe multi-view DSVLs, such as the UML or VisMODLE formalism to which this work directly contributed [19]. The XML User Interface Language (XUL) is used to represent appearance and visual interfaces. It is a language derived from XML that describes user interfaces. XUL is not yet a public standard, but it uses many existing standards and technologies which makes it easily readable for people with a background in web programming and design. The main benefit of XUL is that it provides a simple definition of common user interface elements (widgets). This drastically reduces the software development effort required for visual interfaces, which has represented the basic motivation for interpreting it in the CRADLE framework.

These are notations made of a set of different diagram types, each one describing a different aspect or viewpoint of the system and are suitable for the future enhancements of our framework.

References

1. Thomas R. Kochtanek and Karen K. Hein: Delphi study of digital libraries, Inf. Process. Manage. **35** (1999), no. 3, 245-254.
2. Edward A. Fox and Gary Marchionini, Toward a worldwide digital library, Commun. ACM **41** (1998), no. 4, 29-32.
3. E. F. Codd, A relational model of data for large shared data banks, Commun. ACM **13** (1970), no. 6, 377-387.
4. Howard R. Turtle and W. Bruce Croft, Evaluation of an inference network-based retrieval model., ACM Trans. Inf. Syst. **9** (1991), no. 3, 187-222.
5. Ricardo A. Baeza-Yates and Berthier A. Ribeiro-Neto, Modern information retrieval, ACM Press / Addison-Wesley, 1999.
6. Dario Lucarella and Antonella Zanzi, A visual retrieval environment for hypermedia information systems., ACM Trans. Inf. Syst. **14** (1996), no. 1, 3-29.
7. N. Reyes-Farfan and J. A. Sanchez, Personal spaces in the context of OA., ACM/IEEE 2003 Joint Conference on Digital Libraries (JCDL 2003), 27-31 May 2003, Houston, Texas, USA, Proceedings. IEEE Computer Society, 2003, pp. 182183.
8. Gonalves, M. A., Fox, E. A., Watson, L. T., and Kipp, N. A. 2004. Streams, structures, spaces, scenarios, societies (5s): A formal model for digital libraries. ACM Trans. Inf. Syst. **22**, 2 (Apr. 2004).
9. Martin Gogolla, Rudolf Herzig, Stefan Conrad, Grit Denker, and Nikolaos Vlachantonis, Integrating the er approach in an oo environment., ER, 1993, pp. 376-389.
10. Claudio Bettini, Review - temporal entity-relationship models - a survey., ACM SIGMOD Digital Review 2 (2000).
11. Birol Berkem, Aligning it with the changes using the goal-driven development for uml and mda., Journal of Object Technology **4** (2005), no. 5, 49-65.
12. J.W. Reeves, What is software design?- available at bleading-edge.com, 1992.
13. de Lara J., Vangheluwe H. AToM3: A Tool for Multi-formalism and Meta-modelling. FASE 2002: 174-188.
14. *www.mozilla.org/projects/xul/*.
15. *xdoclet.sourceforge.net*.
16. Robert N. Oddy, Stephen E. Robertson, C. J. van Rijsbergen, and P. W. Williams (eds.), Information retrieval research, proc. joint acm/bcs symposium in information storage and retrieval, cambridge, june 1980, Butterworths, 1981.
17. David Ellis, The dilemma of measurement in information retrieval research., JASIS **47** (1996), no. 1, 23-36.
18. S. Levialdi, A. Malizia, "Modeling Collaborative Interactions in a network of Digital Libraries", Proceedings of the I International Conference on Multidisciplinary Information Sciences and Technologies (INSCIT 2006), Merida (Spain), Volumen I, 181-185, October 2006.
19. Alessio Malizia, Esther Guerra, and Juan de Lara, Model-driven development of digital libraries: Generating the user interface, Proceedings of the ACM/IEEE 9th International Conference on Model Driven Engineering Languages and Systems (MD-DAUI 2006 workshop), Vol-214, ISSN 1613-0073, CEUR, 2006.