MANAGING THE DEVELOPMENT OF DIGITAL LIBRARIES, AIDED BY A TOOL BASED ON 5S

Sai Tulasi Neppali

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE
in
Computer Science and Applications

Edward A. Fox, Chair
Richard Furuta
John Impagliazzo
Clifford A. Shaffer

August 2, 2012
Blacksburg, Virginia

Keywords: Digital Libraries, 5S, Project Management

MANAGING THE DEVELOPMENT OF DIGITAL LIBRARIES, AIDED BY A TOOL BASED ON 5S

SAI TULASI NEPPALI

Abstract

An increase in collaboration between organizations building digital libraries has strengthened the need for tailored project management systems. Current project management tools do not handle the unique challenges that digital libraries pose from a management perspective. They require significant time to apply, and only provide partial solutions. To address this issue, we have designed, implemented, and deployed a Web-based management solution which caters to the planning needs of the project team during the initial phases of digital library development. The main aim of the tool is to capture digital library development patterns and to make development easy for collaborators. The key aspects of the tool are: i) use of the 5S framework of digital libraries to describe the managerial information system as well as the digital library being managed itself, ii) design that is centered on keeping the user informed about digital library practices by integrating appropriate user affordances into the tool, and iii) providing generic project management features such as timeline-creation and task-scheduling in addition to digital library customizations. A formative evaluation of the tool was performed using CTRnet and a national digital library as case studies.

Acknowledgements

I would like to thank my advisor and committee chair Dr. Edward A. Fox for his support and guidance throughout this endeavor. Dr. Fox has been my mentor in many ways and a role model whose qualities I much admire and hope to emulate.

I also would like to thank my other committee members, Dr. Richard Furuta, Dr. John Impagliazzo and Dr. Clifford Shaffer for being a part of the defense committee and for their valuable comments and suggestions.

I thank the members of the Qatar digital library team for being the inspiration behind this work. I thank Kiran Chitturi and Tarek Kanan for helping me with the evaluation of the tool.

I thank the members of the Digital Library Research Laboratory for being an extremely hardworking and down-to-earth group of people who made the lab a comfortable place to work in. Fang Jin's enthusiasm is contagious and I thank her for the many times she came over to strike-up a conversation or to share her food.

My biggest thanks go to my family, especially my mother who has made innumerable sacrifices to see me get a good education.

I dedicate my work at feet of my lord, Bhagwan Baba.

This material is based in part upon work supported by the National Science Foundation under Grant Nos. IIS - 0916733 and 1319578. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation. Assistance also was provided by personnel working on Qatar National Research Fund Project No. NPRP 4-029-1-007.

Table of Contents

List of Tables

List of Figures

1. Introduction

1.1. Context

1.1.1. What is a Digital Library?

The concept of a digital library has roots in Vannevar Bush's seminal work "*As we may think,*" where he wrote about the use of *memex,* a hypothetical proto-hypertext system for information retrieval applications [1]. The vision of a computer-based library spread with J.C.R. Licklider's work in 1965 [2]. Although known by many names, such as virtual library or electronic library, the term "digital library" became the standard terminology propagated by the Digital Libraries Initiative (DLI) in 1993. The National Science Foundation (NSF) supported federal programs DLI and DLI2 which involved institutions such as libraries, universities, associations, corporations, foundations, and various government agencies with substantial funding. At the time, this initiative was the single most visible effort to develop the digital library concept [3].

The Digital Library Federation, a program of the Council on Library and Information Resources (CLIR), defines digital libraries as "organizations that provide the resources, including the specialized staff, to select, structure, offer intellectual access to, interpret, distribute, preserve the integrity of, and ensure the persistence over time of collections of digital works so that they are readily and economically available for use by a defined community or set of communities" [4]. To document a common understanding of the term "digital library" and to outline its scope, the DLib Working Group on Digital Library Metrics defined a digital library as a "collection of services and a collection of information objects that support users in dealing with information objects and the organization and presentation of those objects available directly or indirectly via electronic/digital means" [5]. Additional definitions of digital libraries from different perspectives are presented in Section 2.1.

1.1.2. Digital Libraries and Project Management as Information Systems

An information system is a combination of hardware, software, infrastructure, and people that together support planning, decision–making and management activities [6]. Digital libraries are one of the most advanced and comprehensive information systems involving components such as retrieval tools, including indexers and classifiers interacting with different classes of people, from expert researchers to novice users [7]. Understanding how such multifaceted systems are planned and managed requires an understanding of another type of information system—the Project Management Information System (PMIS) and also project management in general. The digital library information system and the project management information system are elaborated in Chapters 4 and 5.

Project management is the application of knowledge, skills, tools and techniques to a broad range of activities for the purpose of meeting the requirements of a particular project [8]. In the five stages of project management, namely: initiating, planning, executing, controlling and closing, the first two, in the context of digital libraries, involve the processes by which the project's aims, services and products are identified. The five stages provide the context and rationale behind the digital library, and they identify how the digital library will benefit the organization for which it is being undertaken. Planning promotes the sustainability of a digital library project [9]. Planning a digital library involves task-scheduling as well as event and resource management. Planning is also important in light of an increase in digital library collaboration in recent years due to factors such as lack of funds in the digital library community, and the international scale of demand for services and information [10]. Sections 1.2—1.8 elaborate on the role of project management in digital library development and discuss the foundations for this thesis.

1.2. Motivation

While there are many software project management solutions, there is currently no open-source tool building effort that is concentrated towards managing digital libraries in particular. The planning stage of digital libraries involves making decisions regarding scheduling, task allocation and other activities such as establishment of communication channels among stakeholders, and making decisions regarding critical aspects of the digital library.

Planning digital library development can be thought of as a foundational exercise since a well-specified organization of the relationships between tasks, phases and people helps to bring the digital library to life within time and budget constraints.

Considering digital libraries from a management perspective, two important observations can be made. These observations provide the motivation for this work. They are stated as:

- The life-cycle of development, from conception to deployment, involves processes such as collection development, metadata creation and content-classification. Regardless of the size, content-type or nature of the digital library, they follow this specific pattern of development. There is no project management tool, currently, that utilizes this information to tailor their services for digital libraries in particular.

- A model-based approach to designing a digital library can describe: i) the types of multimedia information it supports (Streams Model), ii) how that information is organized (Structures Model), iii) the presentational properties and operations of its components (Spaces Model), iv) the services provided by the digital library (Scenarios Model), and v) the different classes of actors that collaborate to build or work with the digital library (Societies Model). These notions have been organized and formalized into the 5S (Streams, Structures, Spaces, Scenarios, Societies) formal framework [11]. 5S has been proven to effectively describe a digital library's abstractions [12]. However, 5S has a broader descriptive power and no previous work has used 5S to guide the development of a project management information system.

Existing project management systems that are popular in the digital library community include Microsoft Project, Basecamp and Scholar. Section 2.5 explores two of these solutions in the light of digital libraries. Some of these tools are commercially available but none is specifically geared towards digital library management. This work describes a project management tool for digital libraries using 5S, and how it was implemented as an online Web application.

1.3. Problem Statement

Digital libraries are one of the most advanced and widely-used types of information systems. Almost all publishers and scholarly societies have their own digital libraries (e.g., multi-national-level, and university-level DLs). Despite there being many well-established digital libraries, no serious effort has been made to study their construction, including collaboration, staffing and project management issues. This situation stands in contrast to the strides made in areas such as digitization and metadata extraction [13]. Until now, only generic project management tools like Microsoft Project or Redmine have been used to develop project plans. These solutions are lacking in some areas of management, such as policy management, that are integral to digital libraries. It is more helpful to plan a digital library by using a project management tool that is embedded in the digital library context rather a generic one. Sections 1.4—1.7 throw light on the proposed solutions, and on the approach adopted to develop such a project management tool.

1.4. Hypotheses

The 5S framework has been used to describe digital libraries, but it is likely to have a broader utility, and be capable of modeling any information system. Since it is definitional and not oriented to any particular programming language, it can lead to implementations in any programming language [7].

The hypotheses of this work are:

- The 5S framework can be applied to describe both CTRnet [14] and a national digital library leading to one general metamodel in an extended version of 5SL, i.e., a suitable XML schema.

- The 5S framework also can be extended and applied to describe a digital library planning information system (for the development and maintenance phases of DL deployment) leading to a general metamodel in an extended version of 5SL.

- A user interface can be devised, based on these two metamodels, to capture both the descriptions of the digital library planning activity, and the planned digital library, that is usable by and helpful to planners and developers of CTRnet and a national digital library (i.e., the UI is helpful for these two efforts, and a formative evaluation shows that only a small amount of additional change is required).

1.5. Research Questions

The research questions that this thesis hopes to answer are:

- Can we build a tool capable of providing project management services that are tailored for digital libraries?
- How can the 5S framework be used to model the digital library development processes as well as the digital library information system?
- Will the tool be able to help researchers and users to perform better at planning-related tasks in the early stages of digital library development?

1.6. Approach

To address the issues described in Section 1.4 and the questions posed in Section 1.5, we present the development and implementation of a project management tool for digital libraries—the Digital Library Development Management Tool.

The following steps have been undertaken to fulfill this work:

- A literature review of the field was performed to gather information about pre-existing tools that support project management. These tools were examined to identify which generic project management features, such as scheduling and communication-management, the digital library development management tool should include.

- Requirements gathering and analysis were performed to choose features that are desirable in the tool. These features were studied to examine how digital library concepts could be incorporated.

- A review of literature about the 5S framework was performed to describe the tool in those terms.

- The next step was to design presentation slides for the tool's user interface (UI) in order to gain feedback on the placement of UI elements to maximize the ease of use.

- After gaining a working knowledge of Ruby on Rails by consulting books and tutorials, the tool was designed and implemented systematically.

- A formative evaluation of the tool was conducted using case studies of real-world digital libraries. Feedback and suggestions were noted for inclusion in the subsequent changes to improve the tool.

1.7. Research Contributions

This tool makes the following contributions to the field of digital libraries.

- It helps to automate and optimize the management of digital library development.

- It provides a new way of looking at the use of 5S to describe a project management system.
- It provides a dynamic, easy-to-use interface to describe a digital library in terms of 5S.
- It provides a solution for file-management and collaboration issues, and a place for long-term preservation of digital library management information that can be re-used.

1.8. Outline of the Thesis

- Chapter 1 outlines the motivation, problem space, and scope of the research.
- Chapter 2 presents a survey of the relevant literature in the field of Digital Library management and the 5S framework.
- Chapter 3 elaborates on the design and the implementation of the DL development management tool.
- Chapter 4 elaborates on the description of the 5S framework to describe digital libraries.
- Chapter 5 presents the formative evaluation study of the tool.
- Chapter 6 presents conclusions of the work and the recommendations for future work.

2. Literature Review

2.1. Digital Library Definitions

Although there are many definitions of digital libraries, there emerges a common theme amongst the different descriptions. The common points are: [15]

- A digital library is not a single entity;
- A digital library requires technology to link information from many sources;
- The linkages between the many digital libraries and information services are transparent to the end users;
- Universal access to digital libraries and information services is a goal;
- Digital library collections are not limited to documents and they extend to digital artifacts that cannot be represented or distributed in printed formats.

2.2. Digital Library Development

Research in digital libraries has increased significantly over the past decade. In 1999, a time when digital libraries constituted an emerging field, Chowdhury identified the major research areas that fall under the scope of digital library development. These areas of research are still relevant, and have been used by researchers as a standard to guide their digital library projects [16]. These fundamental areas serve to identify the phases that constitute digital library development. They are listed in Sections 2.2.1—2.2.10 [15]:

2.2.1. Collection Development

Collection development is one of the most important stages in building a digital library. Formulating collection development policies, apart from serving as rules for content-selection, steer the team into closely examining the project's aims and prioritizing them.

2.2.2. Development Methodology and Design Issues

There are several ways of modeling a digital library's architecture. This area discusses what kind of strategy is followed while developing the digital library system. Some sample architectures include a three-tier client-server architecture and a service-oriented architecture.

2.2.3. User Interfaces

User Interfaces are extremely critical components of the “front end” of a digital library. The user interface is responsible for presenting a seamless face of the information in the digital library, regardless of the source or format of the information. For example, there may be many video formats in the digital library’s database, but the user should be sheltered from this variety. Flexibility, accessibility and coherence are central user interface issues.

2.2.4. Information Organization: Classification and Indexing

It is important to employ intuitive classification schemes because that is reflected in the browsing and searching services and the UI. Examples of classification systems include the Dewey Decimal System (DDC) and Library of Congress Subject Headings (LCSH).

2.2.5. Resource Discovery: Metadata

Metadata is key information about data. The collection and representation of metadata is determined by the metadata standard. Dublin Core is one of the most widely used standards due to its flexibility and extensibility to include many types of data. One issue, however, may be how the same information from different sources and different formats can be represented in the chosen standard metadata format.

2.2.6. Access and File Management

Access management is of great importance in commercial digital libraries, and it includes policies about user permissions on the information these libraries contain. Bowman and Camarge [45] suggest that integrating file management in digital libraries improves the integration of information, and reduces the time to develop collaborative applications which must be supported by a particular pattern of access.

2.2.7. User Studies

It is important to understand how people seek information in order to tune information retrieval systems according to user requirements. There have been studies in the past [17] illustrating how user-centered design has led to improvement in the design of a digital library component.

2.2.8. Information Retrieval

Users must be able to easily access and retrieve information in a way that it is platform-agnostic. Issues such as indexing, relevance ranking, stemming, stop-word removal, etc. must be considered in this stage of development.

2.2.9. Issues

Digital libraries, being content-driven applications, are exposed to a number of issues, such as legal (copyright-protection, privacy) and social (maintain balance with traditional libraries) issues, and these must be considered carefully and identified during the policy-formulation stage.

2.2.10. Preservation

The goal of a digital library should be to maintain its relevance and provide services for the long-term. This preservation aspect of a digital library must be a theme during all stages of development to guide it in the right direction (e.g., formats of collected data should be based on prevalent standards).

2.3. 5S

5S is a formal model that provides for the theoretical and practical unification of digital libraries [18]. The five constructs in the 5S framework are: Streams, Structures, Spaces, Scenarios and Societies. 5S can describe digital library abstractions such as digital objects, collections, metadata, services, archives, etc. Each of these 5S constructs is described in further detail in Sections 2.3.1— 2.3.5.

2.3.1. Streams

Streams are sequences of elements of any type. They represent static or dynamic content. Static streams constitute the information content of an entity (e.g., text in a document constitutes static content). Dynamic streams represent the flow of information, and are useful in representing the communication that occurs in a digital library (e.g., a video stream or positional data from a GPS on a moving object constitutes dynamic content). Dynamic streams are typically recognized by their temporal nature.

2.3.2. Structures

Structures specify the way in which streams are organized (e.g., a book can be organized into chapters, sections and subsections in a structured format). Markup languages are the most commonly used structures in the Web. They describe the internal structure of digital objects, metadata standards, properties of collections, knowledge organization tools and information retrieval constructs.

2.3.3 Spaces

A space is “a set of objects along with operations on those objects” [19]. Digital libraries can be thought of as being present in the information space. Some examples of spaces include conceptual spaces for the human mind, synthetic spaces for virtual reality systems and document spaces for textual information. In addition, either a 2D or a 3D space typically is used in a digital library GUI.

2.3.4 Scenarios

Scenarios describe interactions between system components in order to deliver services. In the context of the other Ss, they explain what happens to streams, in their spaces, through their structures. They include requirements, operations, services, and tasks that specify the functionality of an information system. Scenarios are useful in the design process of an information system and they help by providing guidelines along which to build prototypes.

2.3.5 Societies

A society is a set of entities along with the relationships between them. Societies constitute not only human stakeholders in the digital library project, but also components such as hardware and software. Societies are the highest level components of the 5S framework, and are a holistic representation of their constituent entities and relationships. Societies include actors, agents, components and modules. Societal issues include those related to economic, legal, organizational, psychological, and other human endeavors as well as the social sciences.

2.4. Management Issues in Digital Libraries

Digital libraries are complex information systems and involve components such as retrieval tools, indexers and classifiers that interact with different classes of people, from expert-researchers to novice users. Managing such systems is, as a result, a formidable task. Hence, project management tools are indispensable parts of the software process. Some of the management issues in digital libraries are:

2.4.1. Collaboration

Michael Hahsler talks about the importance of collaboration in digital libraries, and the increasing need for it [10]. On the other hand, collaboration can pose challenges, some of which include:

- Social issues arising from interactions between project partners

- Economic issues including when coordination leads to confusion instead of allowing for a collective gain

- Technical issues of interoperability between collaborators using different platforms and technologies

- Issues that arise when people from different cultures have different modes of communication, where some prefer email, while some prefer face-to-face talks.

2.4.2 File Management and Sharing

Project management tools should keep up-to-date versions of documents so that all the team members have a clear idea of the status of the project. Solutions like Dropbox are isolated (not integrated with other project management aspects) and expensive.

2.5. Some Existing Project Management Tools and their Features

This section looks at some of the currently-used project management tools for managing digital library development processes, and it describes their advantages and disadvantages as well as highlights the need for a customized digital library management tool.

2.5.1. Microsoft Project

Microsoft Project (MS Project) is a project management software program, developed and sold by Microsoft, which is designed to assist a project manager in developing a plan, assigning resources to tasks, tracking progress, managing the budget, and analyzing workloads [20]. MS Project creates budgets based on assignment work and resource rates. As resources are assigned to tasks and assignment work is estimated, the program calculates the cost, equal to the work times the rate, which rolls up to the task level and then to any summary tasks and finally to the project level. Microsoft Project has particular emphasis on resource-management. Its project management features include timeline creation, budgeting, event management and calendars.

Microsoft Project, however, is a commercial product with a current price of $849. It can be used only on the Windows operating system. Microsoft Project is not designed for software collaboration (i.e., team members cannot use it to communicate tasks and share the editing space). It stands on the local machine and can only be transported as a “Microsoft Project Document” between computers. Exporting it to a more standard exchange format generates an XML file with a complex schema from which it can be difficult to extract specific elements. Microsoft Project does not include a file management system, so file resources cannot be shared between team members. In summary, it is a commercial tool for software management with less emphasis on association between the collaborators of a project.

While many features of Microsoft Project are desirable, it is lacking in the areas of file management and collaboration. Accordingly, the digital library project management tool was designed to avoid these limitations while at the same time include the best features of existing project management tools. Thus, features such as timeline-generation based on task-duration, calendar-use in event management and list-views for large numbers of task entries were included in the new tool. Further, the XML import and export follows a readable and uncomplicated schema.

2.5.2. Redmine

Redmine is a free, open-source, Web-based project management and bug-tracking tool. It includes calendars and Gantt charts to aid visual representation of projects and their deadlines. Redmine provides integrated project management features, issue tracking, and multiple version control options [21].

While Redmine has desirable features like a document management system, it is lacking in project management features like calendars, timelines, events, groups and importing facilities. The user interface has a significant learning curve. In the light of digital libraries, there are entities like policies and events that require emphasis, yet coding them in the form of a plugin for Redmine is cumbersome because of the complex database relationships that come into play between these entities.

3. The Digital Library Development Management Tool: Analysis and Design

Sections 3.1- 3.5 elaborate on the various design decisions made about server-side language, database and user-interface elements in the digital library development management tool.

3.1. Server-side Language: Ruby on Rails

Ruby on Rails (RoR) [22], also called Rails, is a Web application framework developed by David Hansson in 2004. RoR is written in Ruby, a dynamic, interpreted, object-oriented language. Rails is a full-stack framework, meaning that all the layers needed to build a Web application from database support to front-end components are present in it. These layers work together seamlessly using the Model-View-Controller (MVC) pattern where the model is the application object, the view is the display and the controller holds the logic and oversees the way the user-interface responds to user-requests [23]. It is advantageous to use Ruby on Rails for content management systems because of the speed and ease with which one can navigate through the objects. Sites like Groupon and Twitter use Ruby on Rails and the increase in usage is reflected in the statistics of job-growth shown by indeed.com [24]. Keeping all these advantages in mind, Ruby on Rails was used as the framework for the tool. Ruby on Rails currently has 41,696 gems. Gems are Ruby applications or libraries that are analogous to plugins or Java-packages [25]. Many gems like 'formtastic' (for handling forms) [26], Nokogiri (for XML import/export) [27], and 'will_paginate' (for pagination) [28] have been used to build the tool. Figure 1 is a representation of the flow of information from HTML, CSS and Ajax calls in the view to the business logic in the controller and into the database wrapped by the model [29].

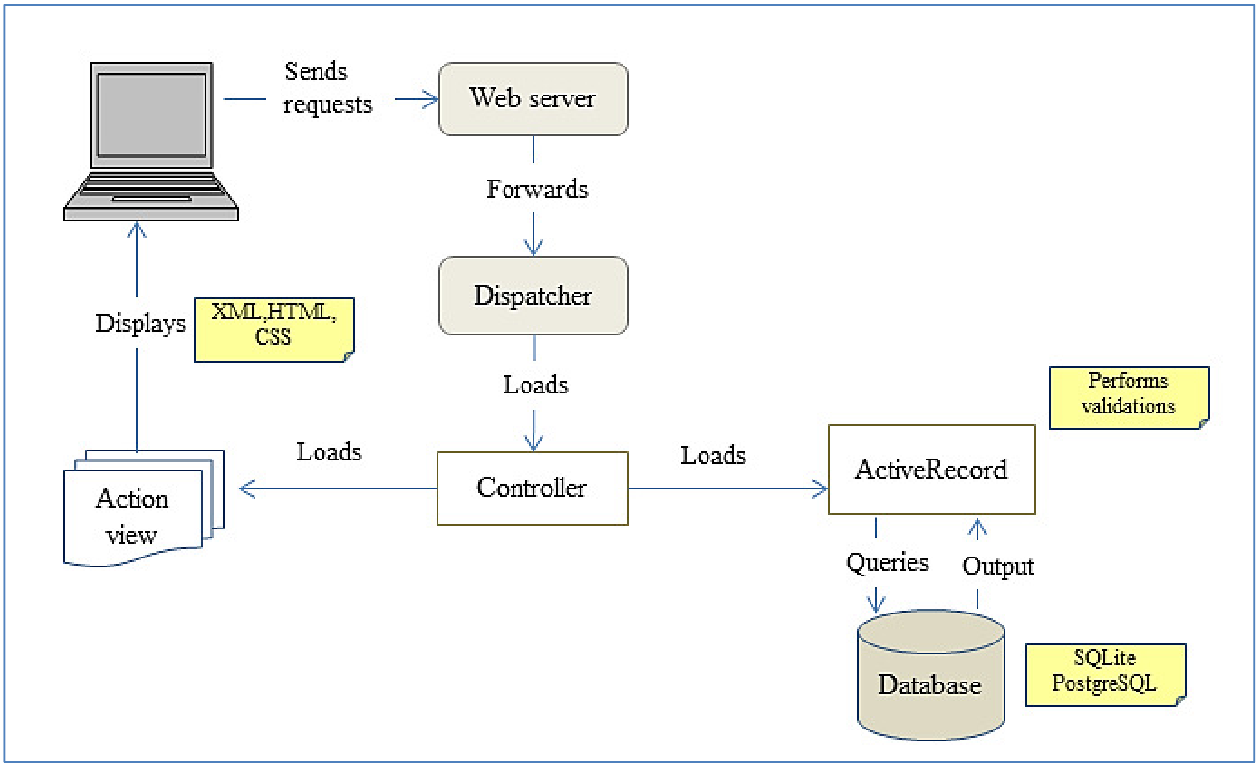


Figure 1. Model-View-Controller pattern in Ruby on Rails

3.1.1. Ruby on Rails Philosophies

Rails' biggest strengths as a framework are the core philosophies that govern the development of its Web applications. Two significant ones are presented in Sections 3.1.1.1 and 3.1.1.2.

3.1.1.1. Don't Repeat Yourself

Every piece of the system should be described once and once only, making development and maintenance much easier. (For example, Rails' views employ partials for pieces of code used repeatedly throughout the application.)

3.1.1.2. Convention over Configuration

Rails follows a naming convention for its models and corresponding controllers and views. This promotes reusability and less code. (For example, if there is a 'Project' model, then the database table is called 'projects'.)

3.1.2. Ruby on Rails: A Comparison with PHP, Java and Python

The debate about the “best” framework / Web programming language is on-going, and one without a definite answer, because no one language or framework is superior in all aspects. There are always trade-offs to be made. However, certain desirable features make a language more suitable in certain scenarios. The issues that are of concern in the context of the digital library development management tool, and those that led to choosing Ruby on Rails as the framework, are:

- Maintainability: The digital library development management tool needs to be sustained over a long period of time to keep it up-to-date with changes in technology. In a talk given by Tim Bray, the director of Web Technologies at Sun Microsystems, he compared the maintainability of Java, PHP and Ruby on Rails, with RoR emerging as a clear frontrunner. Its strict adherence to MVC, templating and Object-Relational Mapping (ORM) are all contributing factors [30].

- Scalability and growth: How many users can the technology support, how quickly can programmers learn the technology to be able to make changes to it as desired and how easy is it to change the layers of the application? These are the questions to ask when considering how scalable a language is. The learning curve is flatter for Ruby on Rails as compared to other Web development languages such as Java or PHP.

- Costs and ease of maintenance: How much does it cost to build and deploy the application, and how easy is it to sustain development, and enhance the application? While PHP has excellent hosting services for websites, Rails is coming up to speed with platforms like Heroku, where hosting a site only takes a small set of commands, and can be streamlined with code on Github [31], a source code version control application.

3.2. Databases with Ruby on Rails

Ruby on Rails supports major database systems like PostgreSQL and MySQL. However, Rails uses ActiveRecord in order to be database-agnostic. ActiveRecord is Rails’ ORM (Object-relational mapping) implementation that follows an architecture pattern of the same name (Active Record). In this pattern, the database table is wrapped into a class, and an object of this class is tied to a single row of the database [32]. Loading and handling of these objects is done by the framework itself through its own syntax, which is converted behind-the-scenes to SQL statements. The biggest advantage of such an approach is that switching between databases is a seamless process. Sections 3.2.1 and 3.2.2 explain the choice of databases the tool currently uses, and Section 3.2.3 elaborates on the database relations in the tool.

3.2.1. Development Database- SQlite3

SQlite3 is shipped as the default development database with Ruby on Rails. It is the most widely deployed SQL database engine in the world [33]. It serves as a fast solution for most common operations and requires zero configuration setup. Its small code footprint, at 350KB, fully configured, and its efficient use of memory, disk space and bandwidth, make it suitable for small to medium-sized websites.

3.2.2. Production Database- PostgreSQL

PostgreSQL is the default database on Heroku, the currently-used application hosting platform for the tool. It has been referred to as the most-advanced open-source object-relational database system in the world [34]. It can support an unlimited database size and a table size of up to 32 TB. PostgreSQL is known to be operationally more reliable than MySQL.

3.2.3. Database Models and Relations

The digital library development management tool has database tables corresponding to the models: User, Project, Phase, Task, Event and Resource. Each of these entities is described in greater detail in Section 4.3. The following are the types of relations that exist between these models:

1) One-to-many: In this relation, rows (records) in one table can be linked to multiple rows in another table, but each row in the second table can only have one linked record in the first table. For example, in the tool, each object in the ‘Project’ table can be related to many objects in the ‘Event’ table through its foreign key project_id, (i.e., each digital library project can include many events, for example seminars or committee meetings.)

2) Many-to-many: In this relation, rows (records) in one table have links to multiple rows in the second table and vice versa. Rails provides two ActiveRecord associations for implementing this relation.

   a. has_and_belongs_to_many: Provides a simple lookup table to reference the two associated models.

   b. has_many :through: Provides a join model which can have columns other than the primary keys of the participating tables.

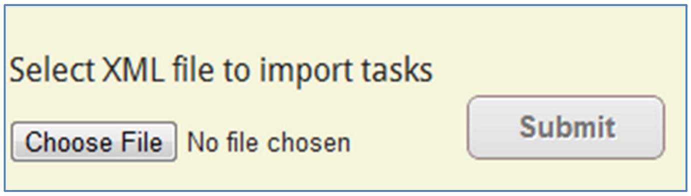
The has_many :through association was chosen for the many-to-many relation between the Project and the User models (a user belongs to many projects and a single project can have many users) because the relationship needs to be treated as an independent entity that can provide validations and callbacks. The join model was called Delegation and it has the column "role". It also has the accompanying "responsibility" apart from the "project_id" and "user_id" since a user's role can vary from one project to the other.

3.3. Ruby Gems

Gems in Ruby are packaged libraries that are analogous to Java's libraries. There are currently 40,000+ gems hosted on RubyGems.org, the largest collection of gems developed by the Ruby community. Some of the gems used in the tool are elaborated in this section.

3.3.1. Nokogiri

Nokogiri is a fast-performing XML and HTML parser written in Ruby [27]. It has been employed in the tool as the standard parser for importing tasks and phases into a digital library project from an externally created XML file. Figure 2 is a screenshot from the tool that shows the user interface elements that accept an XML file and that performs validation against the XML schema [35, 36] given in Appendix B. If the XML file is properly validated, a success message is displayed, letting the user know that the database has been populated with the XML entries. If the validation fails against the schema, an error message is displayed, giving feedback about the reason for failure.



Select XML file to import tasks

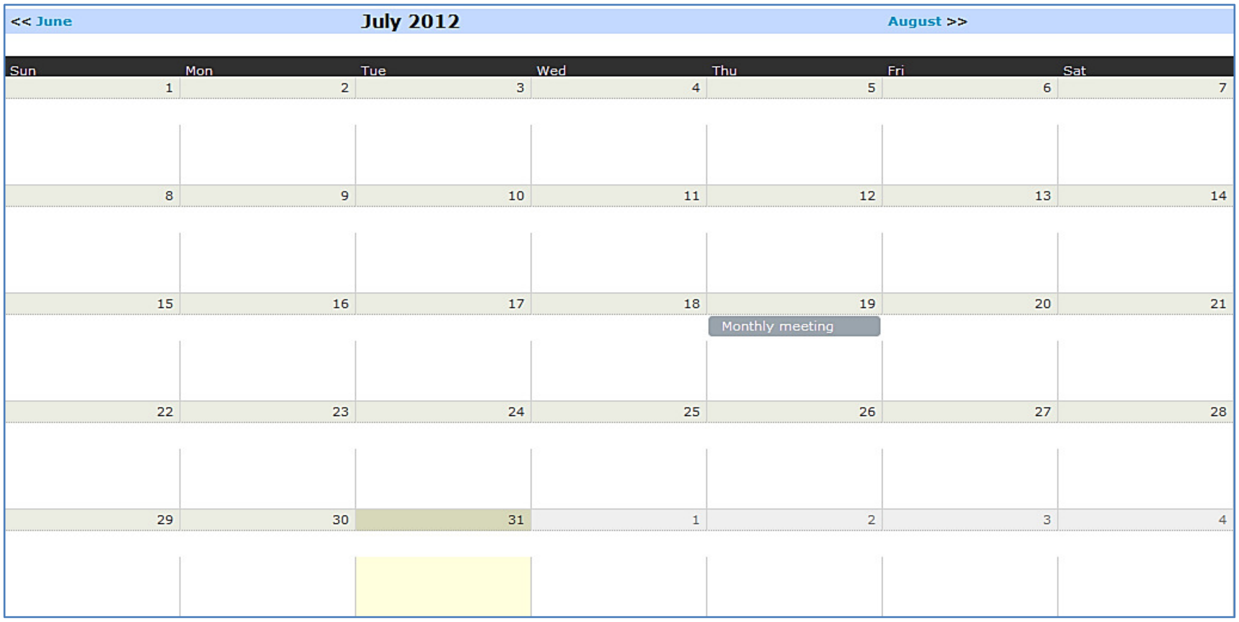
Figure 2. Import tasks in XML format

3.3.2. Will_paginate

Will_paginate is a database layer extension that allows the processing of paginated queries. It is used as a gem to enable the database objects that form the response to be accessed and displayed as a fixed number of objects per page. It also includes view helpers and CSS styling for displaying the page numbers. It is used to display lists of tasks and users in a digital library project.

3.3.3. Event_calendar

The event_calendar gem provides a calendar display of multiple overlapping events. This gem is used to display events in a digital library project, for example, monthly meetings or upcoming workshops. Figure 3 illustrates the event_calendar gem's display in the tool.



<< June July 2012 August >>

Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19 Monthly meeting	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4

Figure 3. Calendar view of events

3.4. JQuery Plugins

Rails supports the JQuery, Prototype and script.aculo.us JavaScript libraries. The JQuery library was used since it has plugins for nearly every type of activity, and a large and vibrant development community. Some of the JQuery plugins used in the tool are:

3.4.1. Simile Timeline

Simile Timeline is open-source software used to create interactive timelines that are also graphically rich. Timelines are used in the tool to represent the phases of a digital library project. Phases are represented as bands spanning from start to end of year (against time on the X-axis). They can be panned infinitely using a mouse pointer. Figure 4 is a representation of the timeline view of phases in a sample digital library.

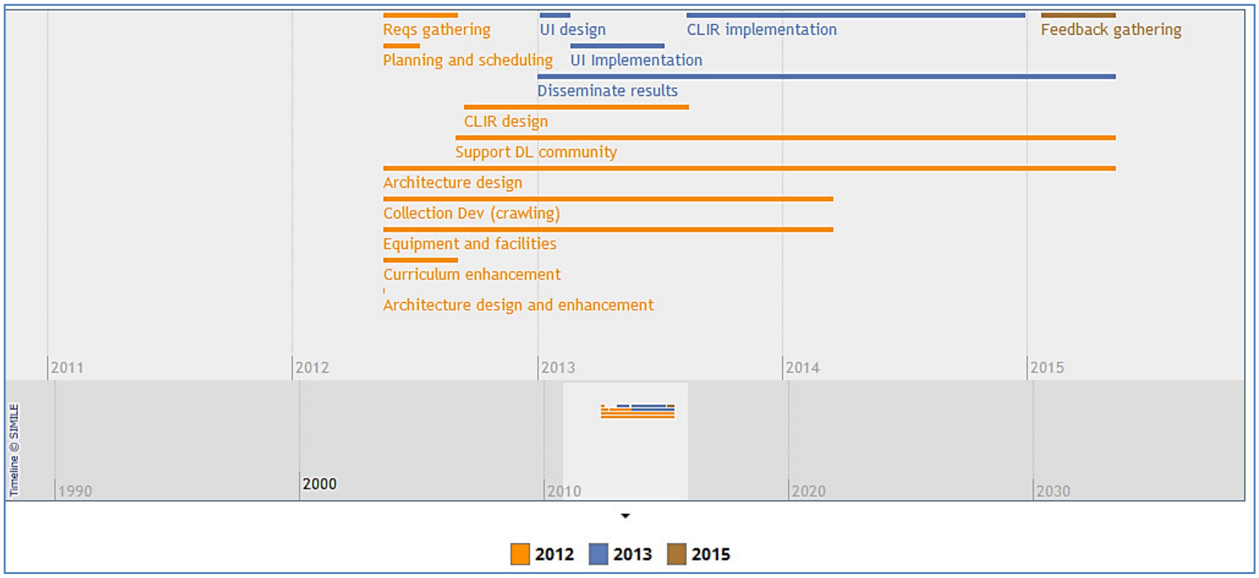


Figure 4. Timeline view of phases

3.4.2. jsUML2/ jsUML2 Editor

jsUML2 is a JavaScript library for visualizing UML2 diagrams that allows embedding a UML2 diagram in an application. The jsUML2 editor built with the jsUML2 library provides an easy-to-use tool for users to create UML diagrams. Of particular interest in digital libraries are sequence diagrams for representing abstractions of scenarios and services (i.e., a set of scenarios) in a digital library. A modified version of the jsUML2 editor was used in the tool. Figure 5 is a screenshot of the toolbar of the editor.

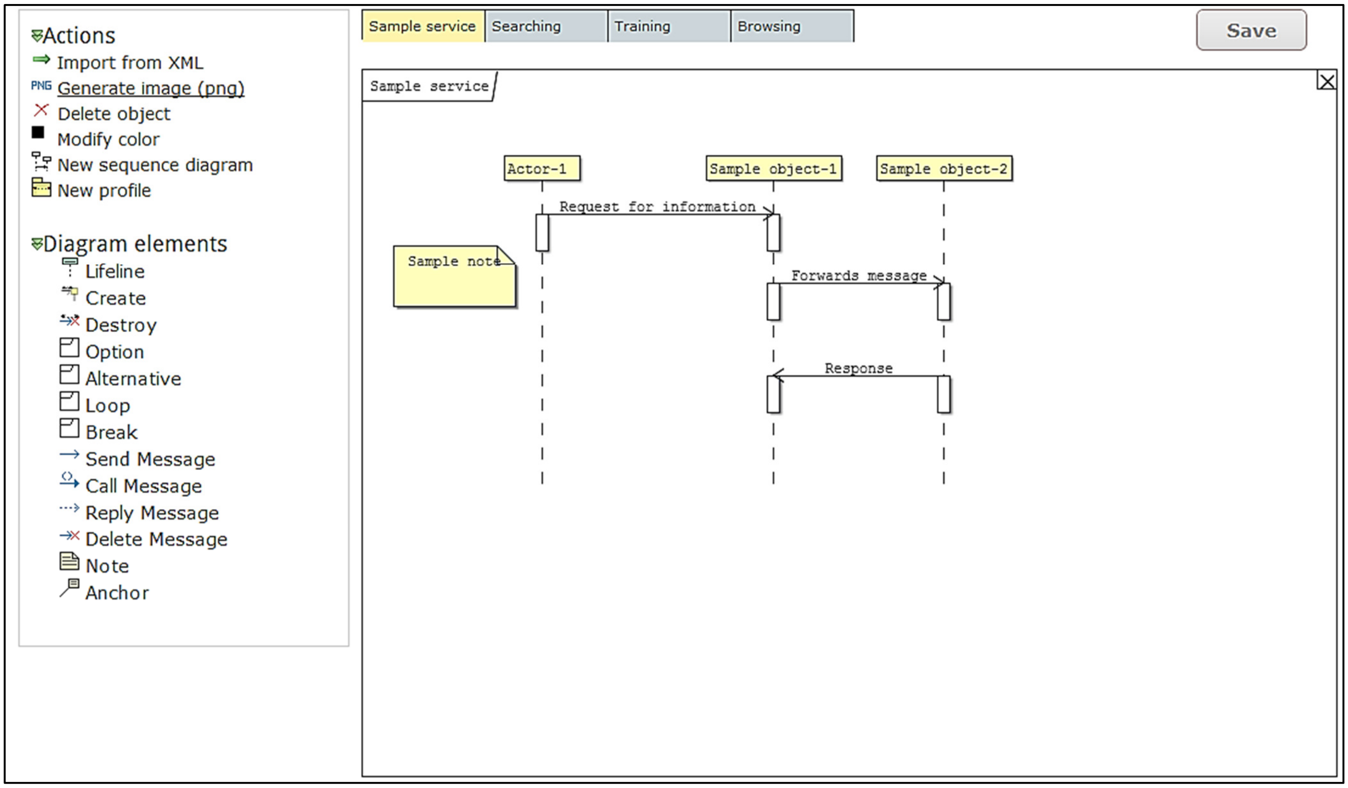


Figure 5. Visualizing digital library scenarios

3.5. User Interface Design

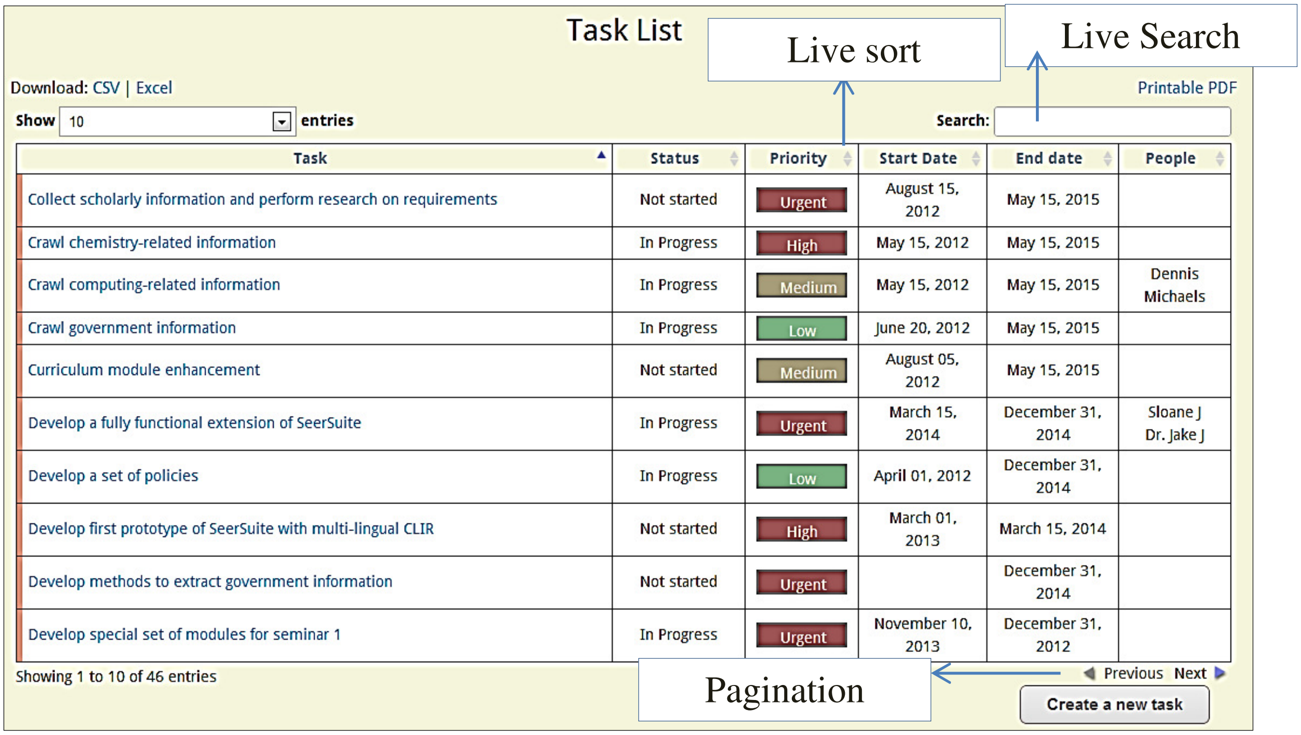
Great emphasis was placed on the user interface of the tool in order to make it usable and appealing to its audience. This section presents the UI design elements that were used in the digital library project management tool.

3.5.1. Layout

The 960 grid system, one of the Web's most widely used grid-based designs, was used as the basis for the layout of the tool [37]. The grid was divided into two-columns, side and main content. This layout provides a large content area, allowing for a greater amount of key information to be presented on a page than that of the 3-column layout.

3.5.2. Ajax Live Search, Sort and Pagination

Providing immediate feedback for an operation performed on an object is an important aspect of user experience. Live searching allows the user to see immediate feedback on the search query, and live sorting performs sorting on any table column without refreshing the page [38]. Pagination is essential when the lists are too long.



Task List

Download: CSV | Excel

Printable PDF

Show 10 entries

Search:

Task	Status	Priority	Start Date	End date	People
Collect scholarly information and perform research on requirements	Not started	Urgent	August 15, 2012	May 15, 2015	
Crawl chemistry-related information	In Progress	High	May 15, 2012	May 15, 2015	
Crawl computing-related information	In Progress	Medium	May 15, 2012	May 15, 2015	Dennis Michaels
Crawl government information	In Progress	Low	June 20, 2012	May 15, 2015	
Curriculum module enhancement	Not started	Medium	August 05, 2012	May 15, 2015	
Develop a fully functional extension of SeerSuite	In Progress	Urgent	March 15, 2014	December 31, 2014	Sloane J Dr. Jake J
Develop a set of policies	In Progress	Low	April 01, 2012	December 31, 2014	
Develop first prototype of SeerSuite with multi-lingual CLIR	Not started	High	March 01, 2013	March 15, 2014	
Develop methods to extract government information	Not started	Urgent		December 31, 2014	
Develop special set of modules for seminar 1	In Progress	Urgent	November 10, 2013	December 31, 2012	

Showing 1 to 10 of 46 entries

Previous Next

Create a new task

Figure 6. Search, sort and pagination in the tool

3.5.3. Usability

To make the tool efficient and easy to interact with, usability was made a priority while designing the tool. The usability principles implemented in the tool are as listed below [39]:

1. Visibility of system status: The tool keeps users informed about its current state through methods such as highlighted user elements and feedback about the user's actions, including those leading to errors as well as to success confirmation messages.

2. User affordances: Matches between the user model and the tool model (digital library and 5S context) are made with the use of hints and other helpful

information about technical terms such as the 5Ss or with the use of prompts for user-input fields in a form.

3. User control and freedom: Users can navigate out of unwanted pages easily by using the navigation links on top of the page or by returning to the home page or overview page of the tool with a single click.

4. Consistency and standards: Standard conventions were followed while naming and placing various user-interface elements like buttons and navigation links.

5. Error prevention: The tool uses messages asking the user to give confirmation for actions such as project deletion that are potentially destructive.

6. Recognition rather than recall: The sidebar in the layout holds a listing of all the currently managed instances for the entity currently being viewed. For example, the ‘phases’ sidebar holds the list of the phases the digital library development will go through and this information is useful when the user changes or adds new phases to the project. This feature ensures that users can depend on the tool to display the objects and options available, and that they need not commit this information to memory.

7. Minimalist design: The tool’s design tries to avoid overloading the user with content, and rather focuses on displaying only as much information as is needed. The features of the project management tool are also kept at a level so as to avoid the phenomenon of “feature creep”. For example, on the 5S structures page of the digital library, collection details are hidden by default using JavaScript collapse/expand tabs.

While the tool adheres to most of the usability principles, there are some issues, such as the lack of a dedicated “Help” section, that are still pending and need to be included in future work on the tool.

4. The Digital Library Development Management Tool: Implementation

4.1. Audience of the Tool

It is important to keep the users in mind when designing an application. The following types of demographics relate to the targeted audience for the tool:

- Research institutions building large-scale digital libraries that require extensive administration and management

- Computer science majors involved in digital library projects

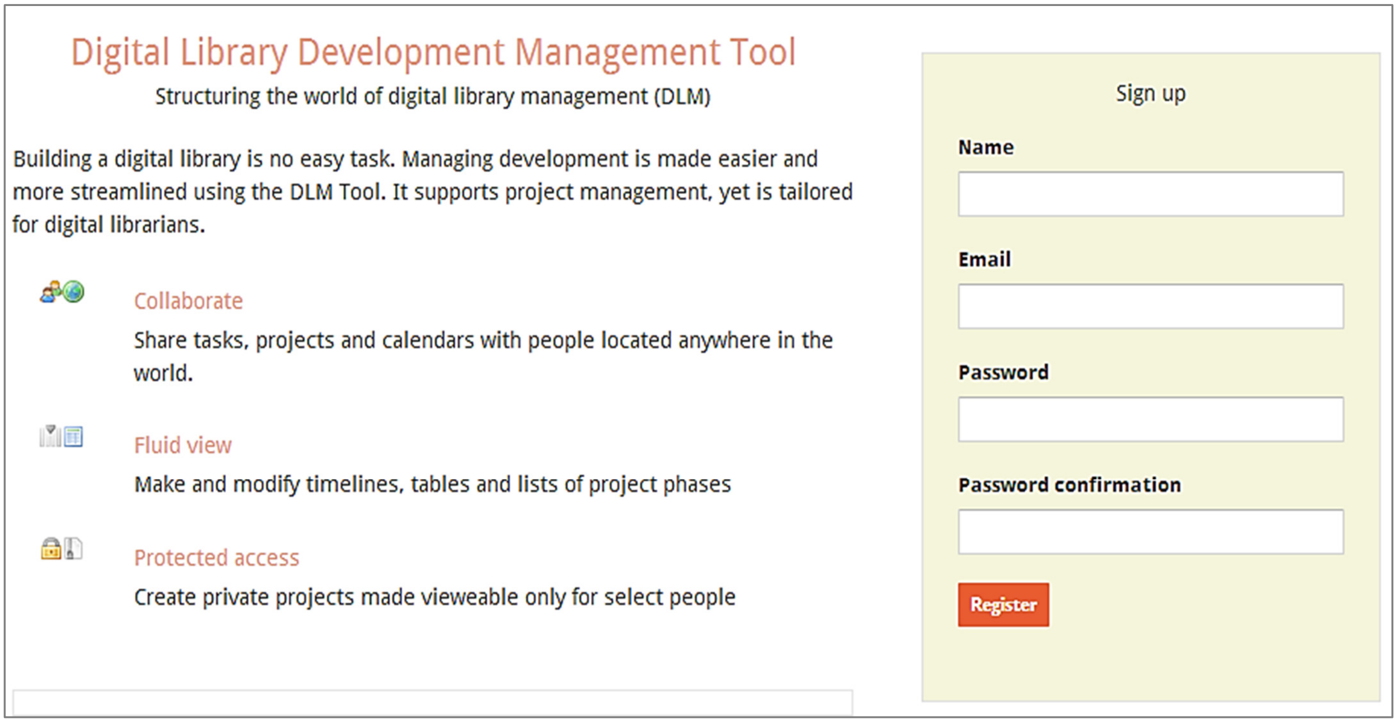
- Librarians involved in the process of converting their collections to digital form

4.2. Features of the Tool

4.2.1. Web-based Access

The digital library development management tool is online and resides on a Web-server. Its primary advantage as a project management tool arises from being available to users 24/7 through the Internet, and providing a secure way by which the users can access their digital library management projects and collaborate with others.

Users can sign-up to gain access to the tool free-of-charge and create and manage their digital library projects. Figure 7 displays the sign-up page of the tool. Users also can invite others to sign-up using the “Invite people” button in their profile page. This feature is useful to invite team members in a digital library project to join and collaborate online.



Digital Library Development Management Tool

Structuring the world of digital library management (DLM)

Building a digital library is no easy task. Managing development is made easier and more streamlined using the DLM Tool. It supports project management, yet is tailored for digital librarians.

Collaborate

Share tasks, projects and calendars with people located anywhere in the world.

Fluid view

Make and modify timelines, tables and lists of project phases

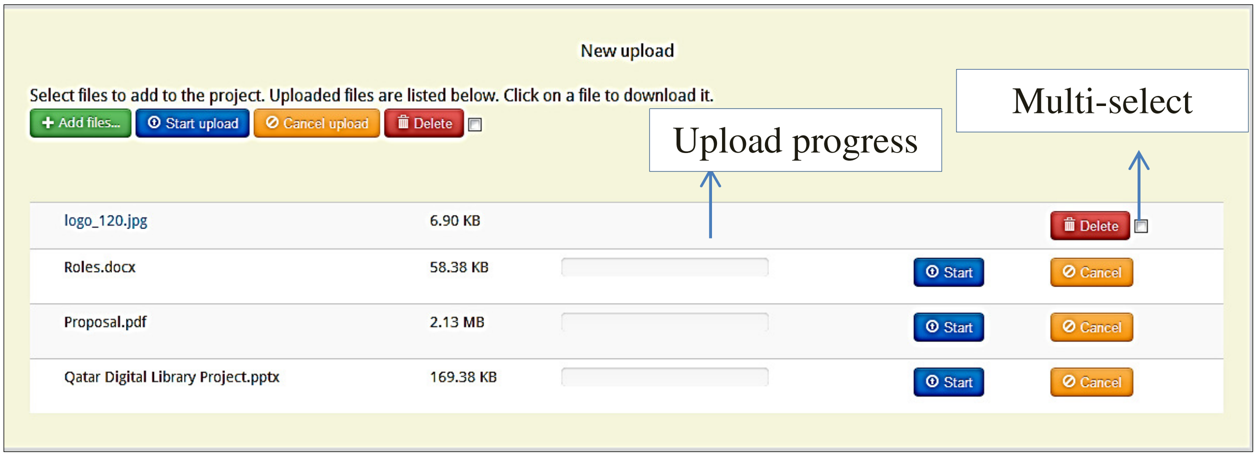
Protected access

Create private projects made vieweable only for select people

Figure 7. Sign-up page of the digital library development management tool

4.2.2. File Management and Sharing

In the planning stages of digital library development, there arises the need to be able to share documents (downloading and uploading) between project users. Documents can include summaries of meetings, questionnaires, schedule lists, project requirement sheets and documents describing peoples' roles. The tool provides an easy-to-use collaboration interface for file-sharing amongst the team members. The screenshot in Figure 8 displays the file-upload feature that can be used to add many files to the project. Error messages, such as about using restricted file extensions, which may arise during the upload, are displayed after a failed attempt. Successfully uploaded files are displayed in a list format; clicking on a file downloads it to the user's local machine. Deleting files is made simple with the use of multi-select checkboxes to remove more than a few files.



New upload

Select files to add to the project. Uploaded files are listed below. Click on a file to download it.

logo_120.jpg	6.90 KB		Delete □
Roles.docx	58.38 KB	Start	Cancel
Proposal.pdf	2.13 MB	Start	Cancel
Qatar Digital Library Project.pptx	169.38 KB	Start	Cancel

Figure 8. Uploading a file to a project in the digital library management tool

4.2.3. XML Import

While the user can manually enter data into the tool one entry at a time, the user also can import data from XML documents which are validated against a specific schema. The advantages of importing data include: i) having more control and flexibility in managing the data, and ii) changes to the imported data will not affect the original data. XML import is available for populating the task, user, and phase tables of the databases. The diagrams representing the XML schema are presented in Figures 9-11 and a complete XML schema is presented in Appendix B. The schema was generated using the oXygen XML Developer tool. The element types include:

a. xs:string – Represents character strings in XML
b. xs:NCName – Represents “non-colonized” names that cannot certain special characters.
c. xs:date – Represents a calendar date
d. xs:anyURI – Represents a Uniform Resource Identifier reference

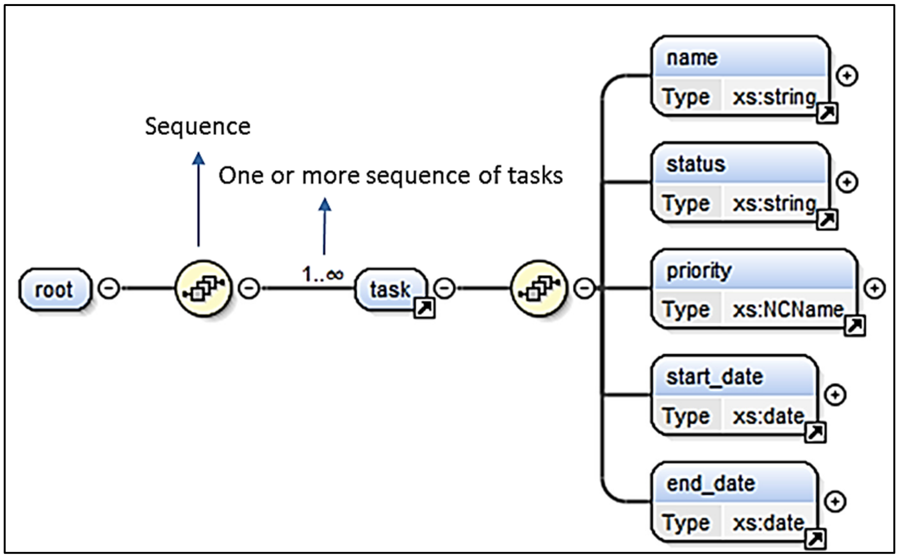


Figure 9. XML schema for tasks in the project

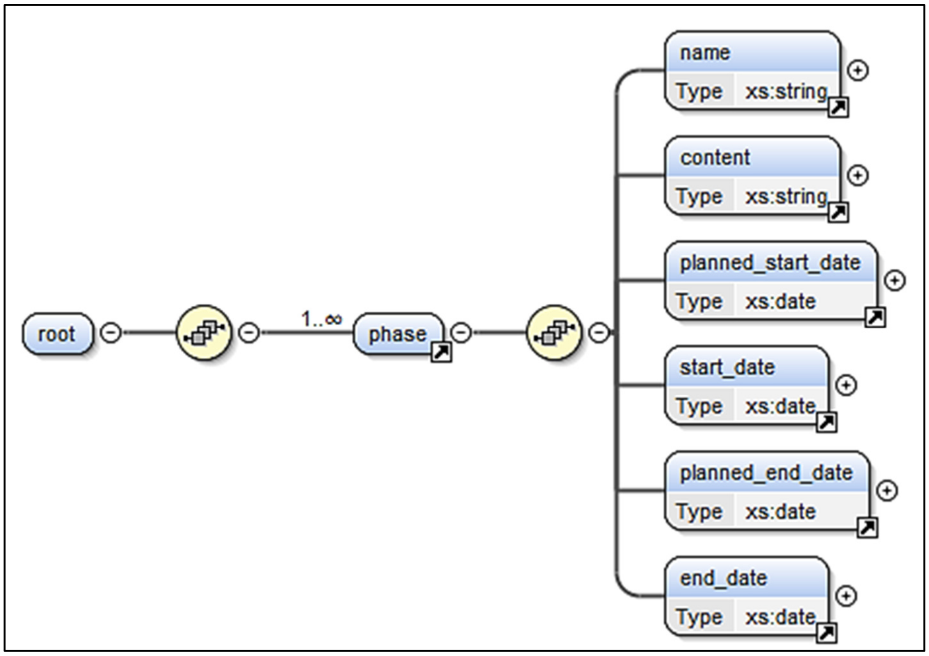


Figure 10. XML schema for phases in the project

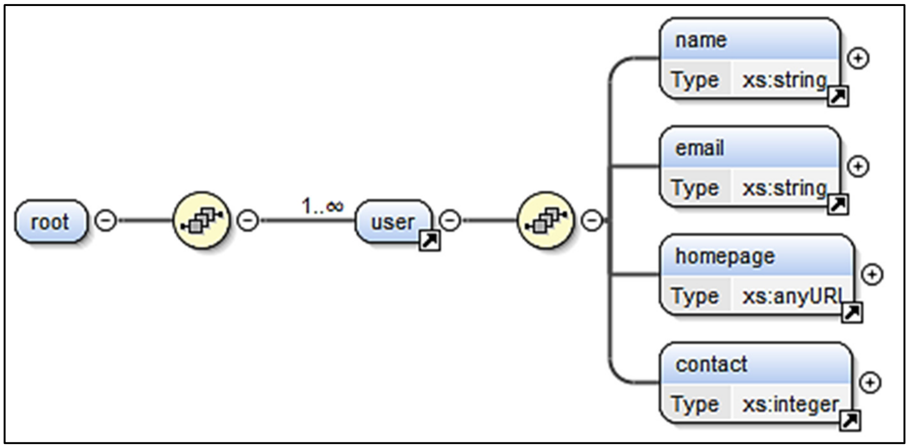
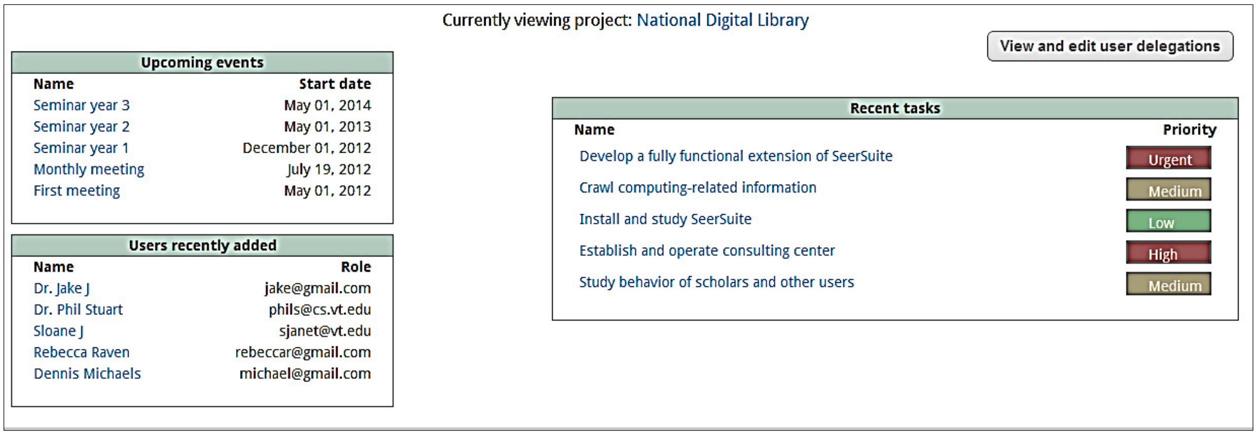


Figure 11. XML schema for users in the project

4.2.4. Dashboard of Recent Updates/Changes

The dashboard screen provides an overview of the projects in which the user is participating, and also provides information about the project currently being viewed. Recently added tasks, project users and upcoming events are shown on the bottom-half of the page. Figure 12 depicts the dashboard page that is a view of the most up-to-date information about the project. Note that the names and addresses shown in the figure are fictitious.



Currently viewing project: National Digital Library

Upcoming events

Name	Start date
Seminar year 3	May 01, 2014
Seminar year 2	May 01, 2013
Seminar year 1	December 01, 2012
Monthly meeting	July 19, 2012
First meeting	May 01, 2012

Users recently added

Name	Role
Dr. Jake J	jake@gmail.com
Dr. Phil Stuart	phils@cs.vt.edu
Sloane J	sjanet@vt.edu
Rebecca Raven	rebeccar@gmail.com
Dennis Michaels	michael@gmail.com

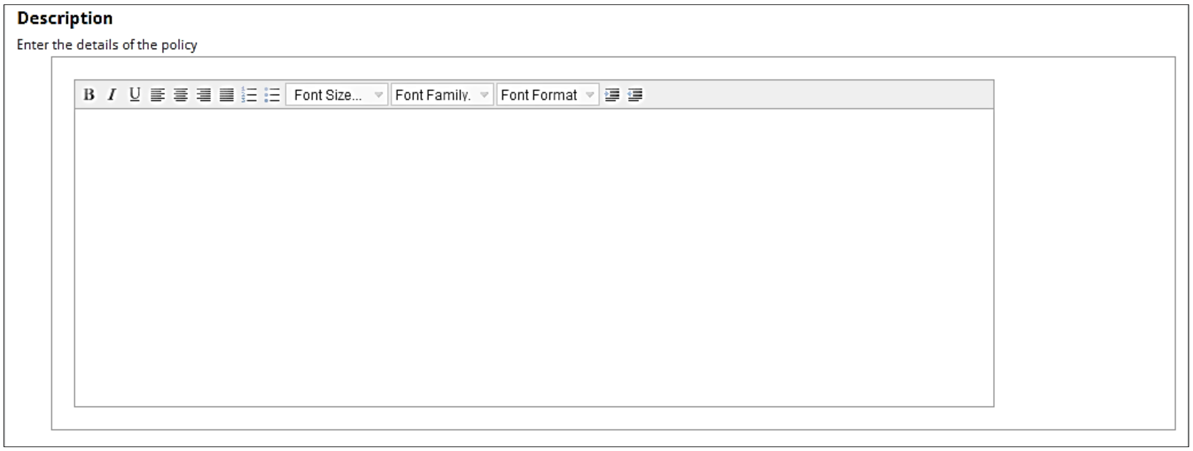
Recent tasks

Name	Priority
Develop a fully functional extension of SeerSuite	Urgent
Crawl computing-related information	Medium
Install and study SeerSuite	Low
Establish and operate consulting center	High
Study behavior of scholars and other users	Medium

Figure 12. Dashboard showing recently added tasks, users and upcoming events

4.2.5. Text Editor and In-place Editing

A user must be able to format information and make changes easily to entries. In-place editing allows the user to correct and change information while viewing its contents. A text editor is used in areas where information needs to be formatted using different font sizes or colors. Figure 13 displays the text-editor used in the tool.


Description

Enter the details of the policy

Figure 13. Text editing

For example, policies consist of rules and regulations that can best be represented by using an appropriate formatting rather than using only plain text. The text editor in this tool has list-formatters, font-formatters and other such standard text-editing functions.

4.2.6. Exporting Data

Exporting data allows it to be used in other applications and promotes interoperability. In this tool the user can export lists of users, tasks, events, policies and phases in three formats:

- CSV (Comma-separated values)
- Microsoft Excel sheet
- Printable PDF file with table-formatting

4.2.7. Searchable Timeline View

One of the available views of the phases of the digital library project is the timeline view depicting the phase-list as horizontal lines along a time-axis based on their actual start and end dates. They appear color-coded by their start-year. Clicking on a phase displays further information about it. The timeline view, as shown in Figure 14, allows the team to envision the high-level structure of the project plan. It is made dynamically searchable so that only phases with names that occur in the search-query string are displayed in the view. This is a highly useful feature when the number of phases is large.

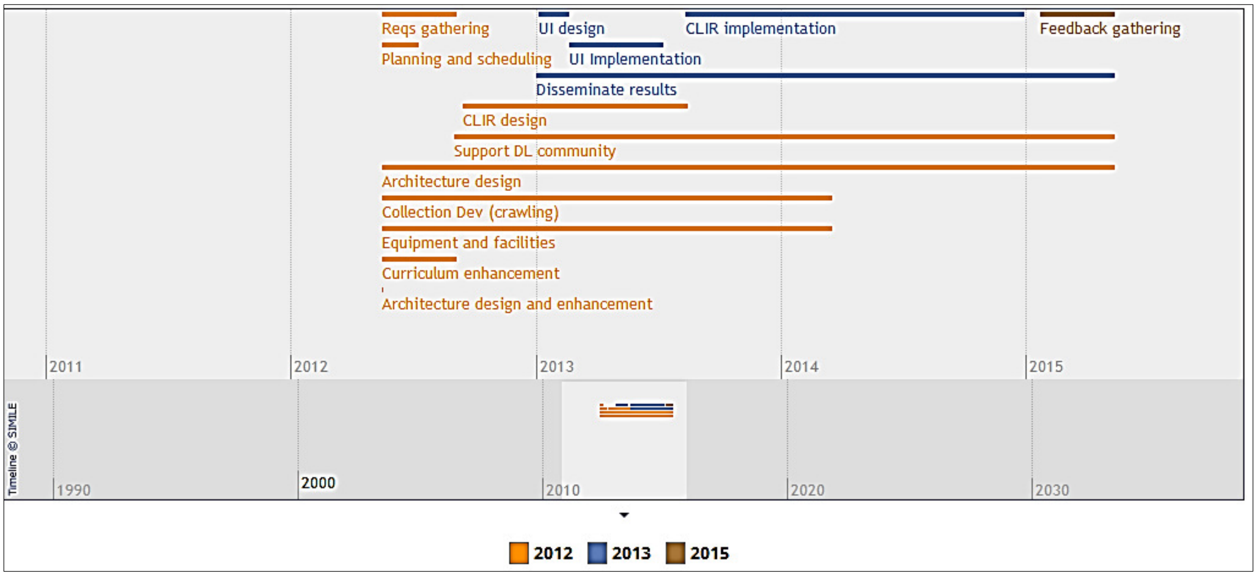


Figure 14. Timeline view of phases

4.2.8. Tree View of Nested Lists

Digital library project phases can be divided into sub-phases with a smaller set of aims. To capture this hierarchical nature of relationships amongst phases, a tree-view representation of the phases has been added to the tool. The view helps identify areas of development that have a greater diversity of tasks than others.

4.2.9. Hints and Guides

The tool has helpful pointers on the user-interface, guiding the user about the nature of the view or the type of information being recorded by the tool. There are hints in the forms that have to be filled out by the user. The 5S framework and description of digital libraries may be new to some of the users. For such users, there are useful hints to guide them through the process.

4.3. Project Management Entities

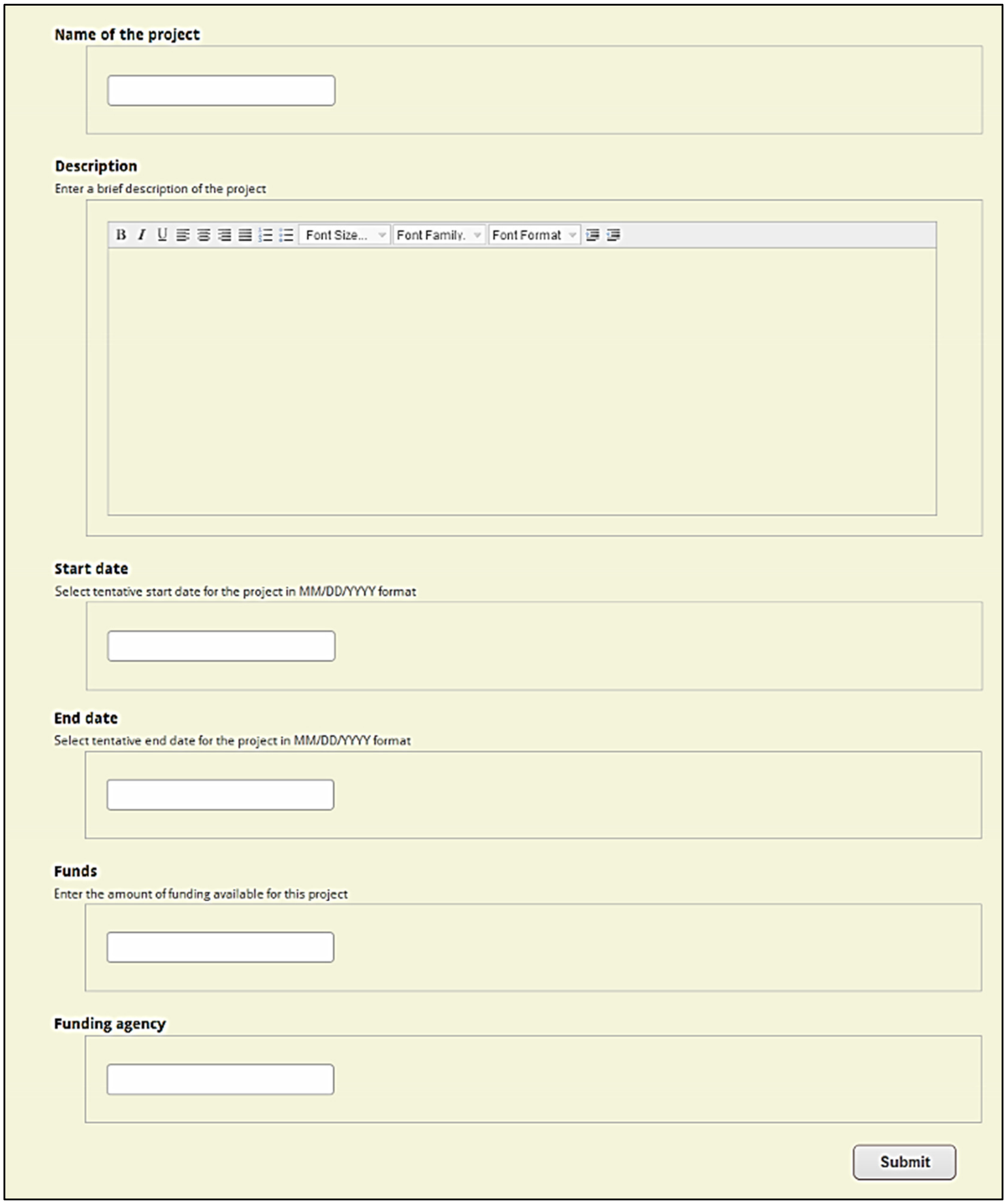
This section describes the building blocks of a digital library development management system and how these entities are represented in the application in the form of database models.

4.3.1. Projects

Projects are the very basis of the tool, and they represent all the information related to a digital library that is being managed using this tool. They can be created, updated or deleted. The project model has database fields, as listed in Table 1, to record project-related information. This table corresponds to the user-interface of the project creation page shown in Figure 15, with fields that perform client-side validations.

Table 1. Database columns of the project model

Column name	Data-type	Description
Name	String	Name of the project/Name of the digital library
Description	String	Notes about the project
Start at	Date-string	MM/DD/YYYY format start date of the project
End at	Date-string	MM/DD/YYYY format tentative end date of the project funding
Money	Floating-point number	Total funding amount allocated to the project
Funding_agency	String	Name of the agency funding the project



Name of the project

Description

Enter a brief description of the project

Start date

Select tentative start date for the project in MM/DD/YYYY format

End date

Select tentative end date for the project in MM/DD/YYYY format

Funds

Enter the amount of funding available for this project

Funding agency

Figure 15. New project creation page

Some project-related scenarios include:

- A user logs in and creates a digital library project by entering project-related information and adds people to the project from the list of registered users.

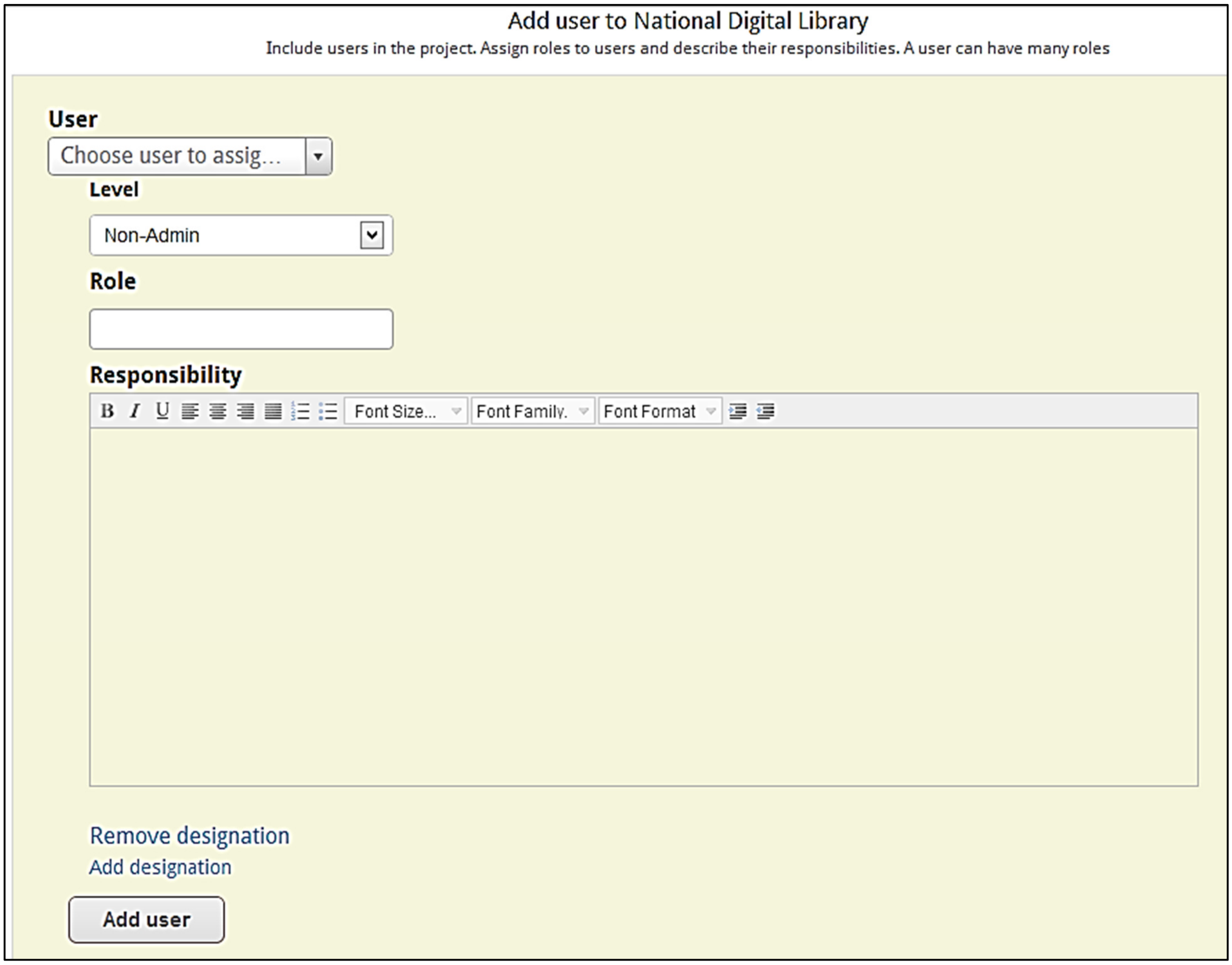
- The newly added project appears in the projects listing page. Clicking on a project sets the data in all of the views to that of the selected project. A user then can continue to add other project entities such as task lists and event schedules by clicking on their respective tabs.

4.3.2. Users

Users of the tool can log into their account with their email and password to access their portfolio of digital library projects. A single user can belong to many projects while a project involves many users. The roles of a user in a project may be more than one, and also they may be different from that in another project of which the user is a part. Users can be assigned tasks after they have been added to the project, as in Figure 16. A description of a project's users (i.e., team members on a project) can be imported from an XML file to populate the database. The database fields of the users table are given in Table 2.

Table 2. Database columns of the user model

Column name	Data-type	Description
Name	String	Full name of the user
Email	String	Email that user needs to log in to the tool
Encrypted_password	String	MD5 hashed password
Salt	String	Randomly generated input for hash encryption
Webpage	String	User's webpage, if any
Number	String	Contact phone of the user



Add user to National Digital Library

Include users in the project. Assign roles to users and describe their responsibilities. A user can have many roles

User

Choose user to assig...

Level

Non-Admin

Role

Responsibility

Figure 16. Add users to the project and assign them roles by selecting them from the user-base.

Some user-related scenarios include:

- A user registers in the tool and logs in to create a digital library project and can involve other users of the tool in that project.

- A user can add tasks, phases, events and policies to a digital library project as well as update or delete them.

- Any user involved in a particular digital library project can edit details about the digital library's 5S framework description.

- A registered user can add users to the project by selecting them from a drop-down list, as shown in Figure 16.

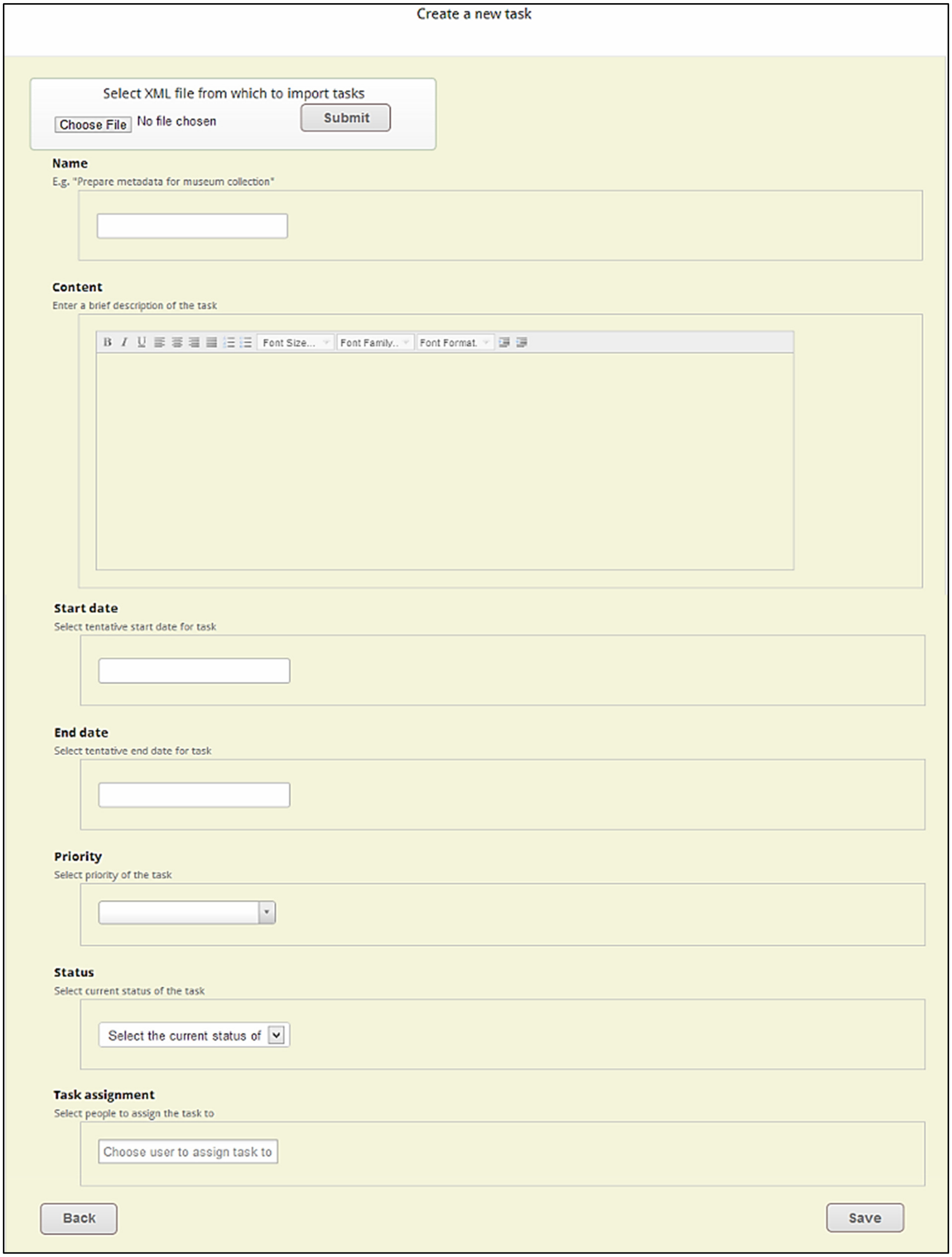
- A registered user also can invite others to sign-up to the tool by using their email address.

4.3.3. Tasks

Tasks are the building blocks of an activity or a management system. Digital library project tasks such as crawling, metadata extraction, classification, etc., all constitute jobs that the project team shares. It is, therefore, important to be able to record task assignments along with time durations, priorities, and settings. The database fields of the users table are given in Table 3, and Figure 17 is a screenshot of the task creation page.

Table 3. Database columns of the task model

Column name	Data-type	Description
Name	String	Name of the task
Content	String	Brief description of the task
Status	Integer	Current state of the task as selected from a list of options
Priority	Integer	Significance of the task as selected from a list of options
Start at	Date-string	YYYY-MM-DD format start date of the task
End at	Date-string	YYYY-MM-DD format tentative end date of the task
Project_id	Integer	Foreign key indicating the project this task is part of



Create a new task

Select XML file from which to import tasks

Choose File No file chosen Submit

Name

E.g. "Prepare metadata for museum collection"

Content

Enter a brief description of the task

Start date

Select tentative start date for task

End date

Select tentative end date for task

Priority

Select priority of the task

Status

Select current status of the task

Select the current status of

Task assignment

Select people to assign the task to

Choose user to assign task to

Back Save

Figure 17. New task creation page

Some task-related scenarios include:

- A project can be divided into tasks that the user can delegate to team-members.

- Tasks can be sorted according to priority so that the most important ones can be shown on top of the list.

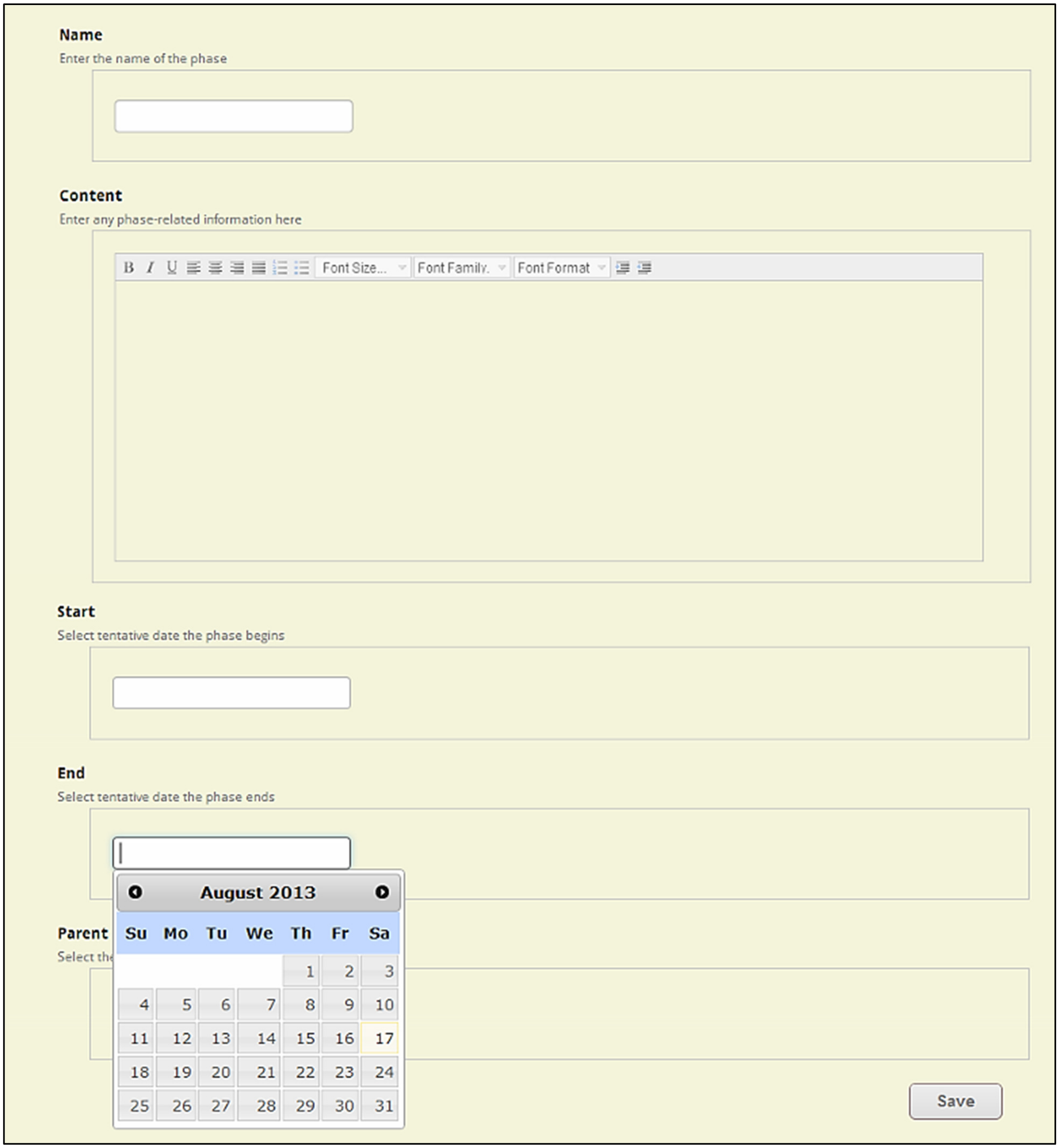
- Tasks are automatically moved into the archives folder for reference purposes once their status is set to "completed".

4.3.4. Phases

Digital library development goes through a set of phases unique and different from other generic software projects. Phases like collection development, indexing, etc., are all part of every DL development process. The project team can, therefore, choose from a list of typical digital library phases to determine what can be included in its digital library project. Custom phases can be created too. The database fields of the phase table are given in Table 4. The corresponding interface screen is illustrated in Figure 18.

Table 4. Database columns of the phase model

Column name	Data-type	Description
Name	String	Name of the phase
Content	String	Brief description of the phase
Parent_id	Integer	ID of the parent phase, if any
Start	Date-string	YYYY-MM-DD format start date of the phase
End	Date-string	YYYY-MM-DD format tentative end date of the phase
Project_id	Integer	Foreign key indicating the project this task is in



Name

Enter the name of the phase

Content

Enter any phase-related information here

Start

Select tentative date the phase begins

End

Select tentative date the phase ends

Parent

Select the

August 2013

Su	Mo	Tu	We	Th	Fr	Sa
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

Figure 18. Phase-creation page

Some phase-related scenarios include:

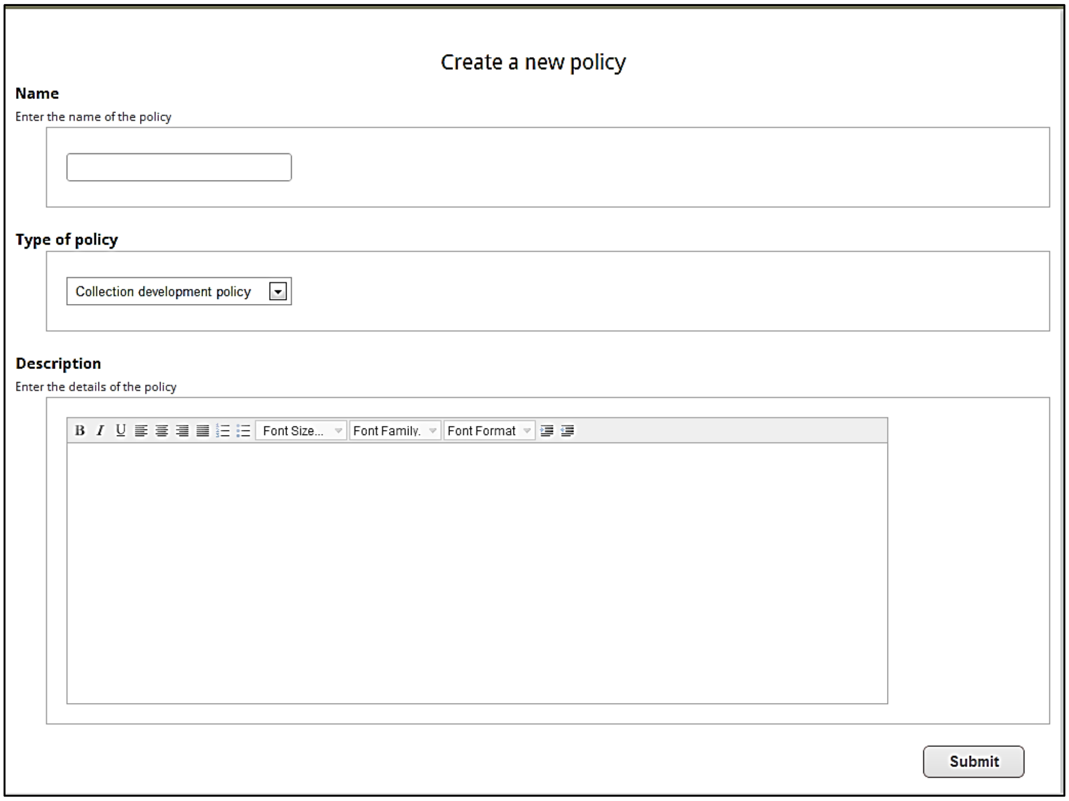
- Project phases can be added by selecting them from pre-defined lists of phases or by creating them by specifying phase-related details as shown in Figure 18.
- Project phases can be nested in other phases, (i.e., can have sub-phases). This nesting is displayed in the form of a hierarchical list of phases and sub-phases.
- Phases can be viewed in a table, list or timeline form.

4.3.5. Policies

Digital library projects adhere to policies that are laid down in the initial planning stages, and are then refined over the course of development. Policies play an important role in defining the scope of the project, the rules by which operations should be performed, and, as such, lay the foundation for the project. Digital library policies typically include those regarding metadata standards, accessibility, privacy, and copyrights, etc. A sample list of such policies is encoded in the tool so that users can choose and take guidance from them while developing their own digital library policies. The database fields of the users table are given in Table 5. Figure 19 illustrates how to specify policies.

Table 5. Database columns of the policy model

Column name	Data-type	Description
Name	String	Name of the policy
Description	String	Policy rules and description
Project_id	Floating point number	Total funding amount allocated to the project



Create a new policy

Name

Enter the name of the policy

Type of policy

Collection development policy

Description

Enter the details of the policy

Figure 19. New policy creation page

Some scenarios involving digital library policies include:

- Digital library policy samples can be viewed in the list format and selected to view sample content. Users then can add their custom policies that are built around a default policy or add one that is not available in the sample-list (as shown in Figure 19).

- Digital library policy samples provided can be used as templates from which custom policies can be created.

4.3.6. Events

Events and meetings are an important part of a digital library project, especially those that involve collaborating partners that meet infrequently. The events model represents these occasions on a calendar, and information about events can be viewed in that particular event's page. A visual view of the duration of the event can be seen on the calendar.

The database fields of the events table are given in Table 6.

Table 6. Database columns of the event model

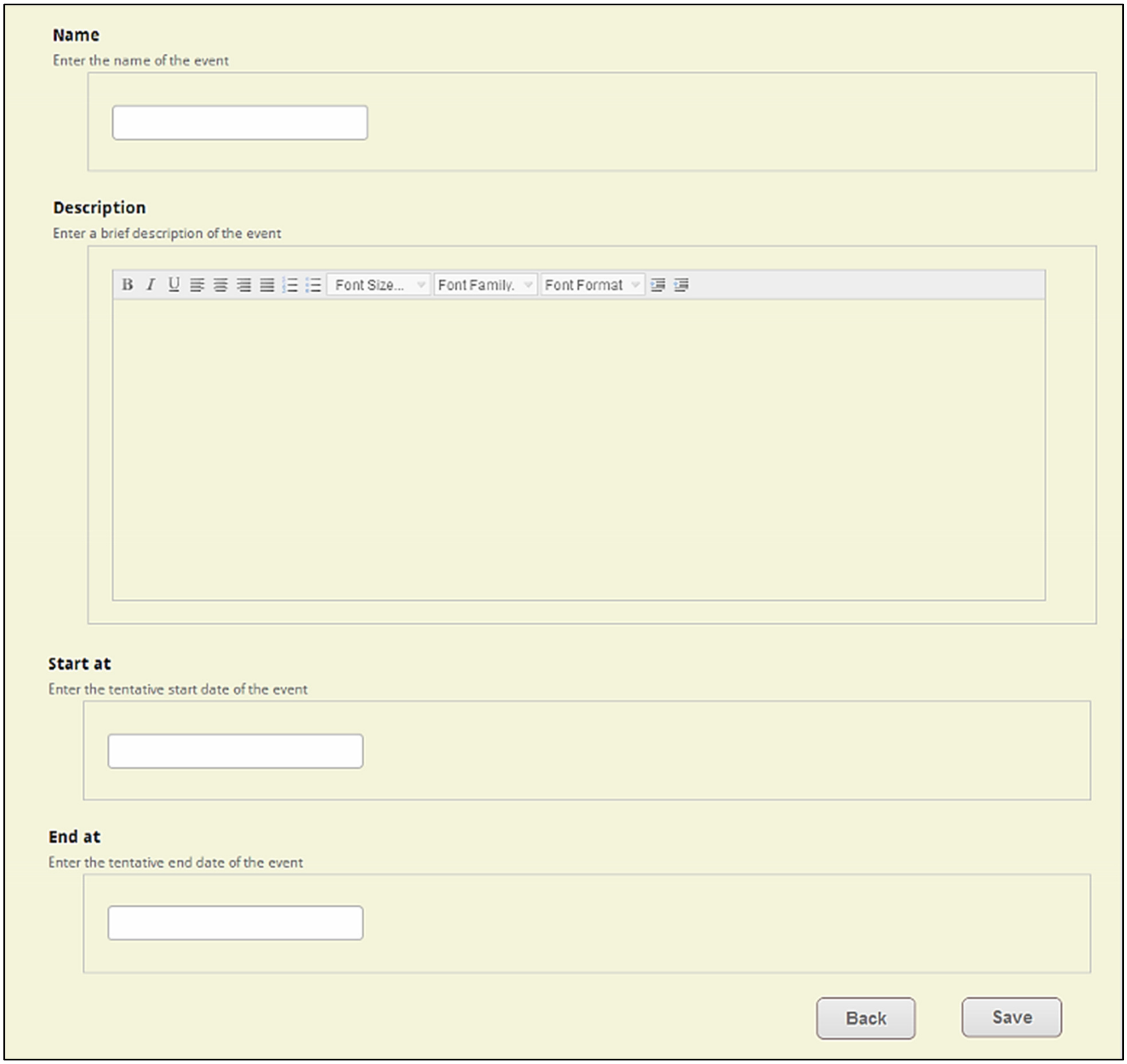
Column name	Data-type	Description
Name	String	Name of the event
Content	String	Brief description of the event
Start_at	Date-string	YYYY-MM-DD format start date of the event
End_at	Date-string	YYYY-MM-DD format tentative end date of the event

Some scenarios involving digital library events include:

- Project events are created and recorded to be viewed on a calendar template. Figure 20 shows the event-creation page in the tool.

- Events are displayed in a tree-like archive format and displayed by month and year of occurrence.

- Event descriptions can be viewed upon clicking on an event.



Name

Enter the name of the event

Description

Enter a brief description of the event

Start at

Enter the tentative start date of the event

End at

Enter the tentative end date of the event

Figure 20. New event creation page

4.4. Database Models in the Tool and their Relationships.

Models represent the data structure of the application. Relationships between models denote dependencies between them that are brought about by joining database tables. The digital library management tool, as described in Section 4.3, has a number of models to handle all the elements such as phases, tasks, and policies that come into light in the formative stages of the digital library. Figure 21 is a diagram of models in the tool along with their relationships with one another.

Active Record associations can be used to describe one-to-one, one-to-many and many-to-many relationships between models. Each model uses an association to describe its role [40]. There are several one-to-many relationships such as Project-tasks, Project-events, Project-policies and Project-resources.

The other important associations in the tool are described in this section.

4.4.1. Project-User

This is a many-to-many relationship. A user can belong to many projects while one digital library project has many participating members. The relationship is manifested as a join model called "Delegation" with fields to represent user "roles" and "responsibilities," since these differ depending on the type of involvement a user has in a specific project. A user can even assume multiple roles within a project. Thus, a user can be a researcher in the team as well as a liaison representing the team to other communities.

4.4.2. Project-Phases

This is a one-to-many relationship with a single project being divided into many phases that are unique to that project. There is also a default list of phases that the user can choose from to include in their project. The 'rgt' and 'lft' columns in the model are used by the nested tree algorithm to determine the parent phase in a nested structure of phases.

4.4.3. Project-Model

This is a one-to-one relationship. The 'Model' in the tool is the 5S description of the digital library. It stores information about the streams, structures, spaces, scenarios and societies of the digital library in an XML format.

4.4.4. User-Task

The join model "Assignment" represents the division of tasks amongst a project's users. It is a relationship between the "Delegation" model and the "Task" model. A single user can be assigned many tasks according to the role that the user plays in the project.

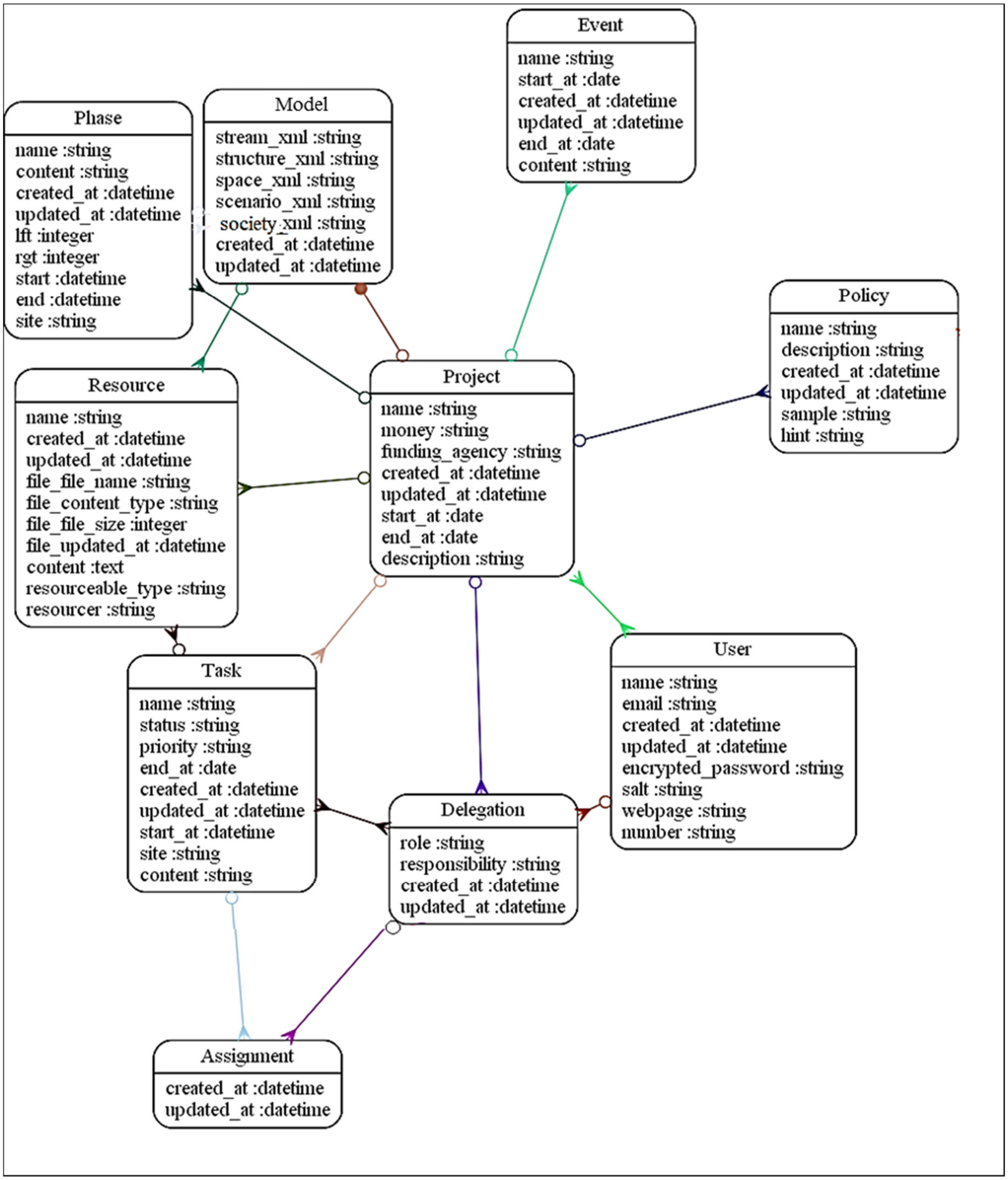


Figure 21. Representation of database models and their relationships

5. 5S Framework and the Project Management Information System

5S, referring to Streams, Structures, Spaces, Scenarios and Societies, can comprehensively represent the components of an information system. Digital libraries in particular have benefitted immensely from the strong theoretical base that 5S provides, including rapid DL specification (5Sgraph [41]), scenario-based DL generation (5SGen [42]), and a quality model tool (5SQual [43]).

5S, as a theoretical framework, allows one to move from simply describing a process to generalizing various aspects of the process [7].

A major contribution of this thesis is that it has shown how 5S can be used to describe a digital library project management information system (see Section 5.2) apart from using it to describe the digital library itself (see Section 5.1). Figure 22 is a diagrammatic representation of this contribution.

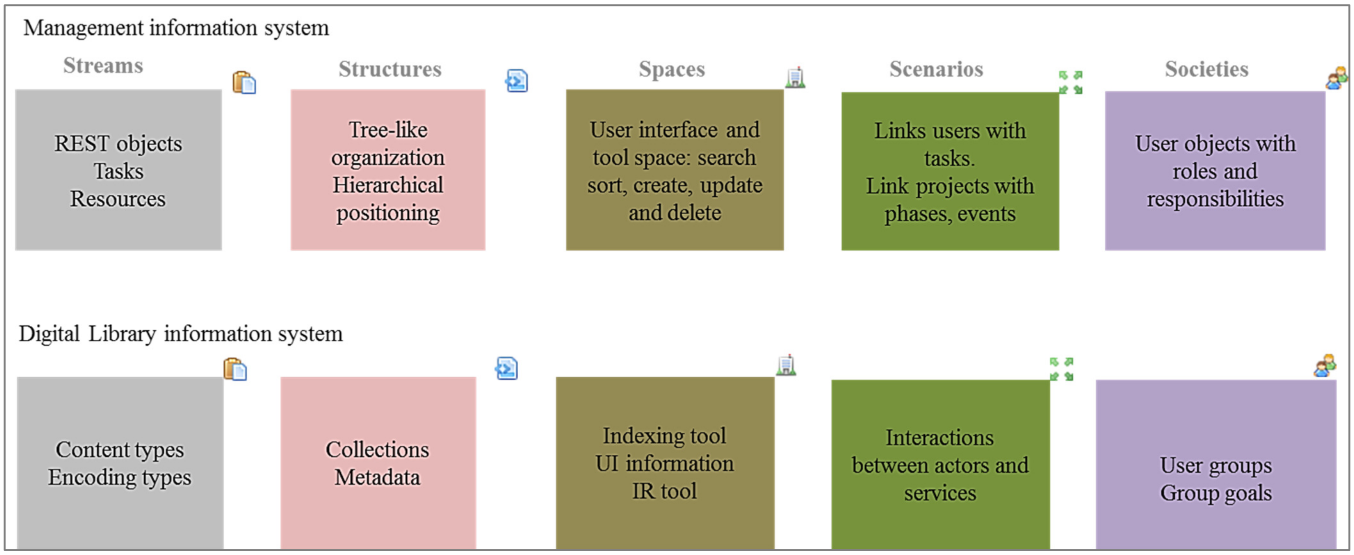

Management information system
Streams
REST objects
Tasks
Resources
Structures
Tree-like organization
Hierarchical positioning
Spaces
User interface and tool space: search sort, create, update and delete
Scenarios
Links users with tasks.
Link projects with phases, events
Societies
User objects with roles and responsibilities
Digital Library information system
Content types
Encoding types
Collections
Metadata
Indexing tool
UI information
IR tool
Interactions between actors and services
User groups
Group goals

Figure 22. 5S used to describe the project management system and the digital library system

5.1. 5S Description of Digital Libraries and the Tool

In this section, we describe and present the implementation of the 5S metamodel used to describe a digital library. Information is a key resource in a project. Capturing information is central from a project management standpoint [44]. It is important for the project leaders to be able to keep track of the decisions made since these are the cornerstones for a project's success or failure. In order to effectively capture information pertaining to digital libraries, it is important for a development team to be able to view and present this information in the right context. Leading in from this rationale, a 5S metamodel for describing the 5S model of a digital library was implemented in the tool. This section elaborates on this implementation in further detail. Table 7 summarizes the various 5S descriptions that are captured by the tool.

Table 7. 5S description of a digital library

5S constituents	**Tool representations**
Streams	Content types Encoding types
Structures	Collections Metadata
Spaces	Indexing tool UI information IR tool
Scenarios	Interactions between actors, use of services
Societies	User groups Group goals

5SGraph is a visual modeling tool that helps to create a digital library instance by capturing its streams, structures, spaces, scenarios and societies models. However, it is a standalone tool that requires updating. It is not integrated with a UML modeling tool and has layout problems [11]. To compensate for the drawbacks of this tool, and with the goal of disseminating the 5S metamodel to the tool's users, the 5S description of digital libraries was added to the new tool.

5.1.1. Describing the Streams Model

The streams model of a digital library specifies the properties of digital information (i.e., the sequence types as well as encodings). Digital content can be represented as text, audio, video or image. 5SL [46] is an XML representation of the 5S model of a digital library. The 5SL representation of the streams in a digital library is presented in the schema in Figure 23. The complete XML schema is presented in Appendix B. Figure 24 illustrates the tool's user interface for describing the streams model of a digital library.

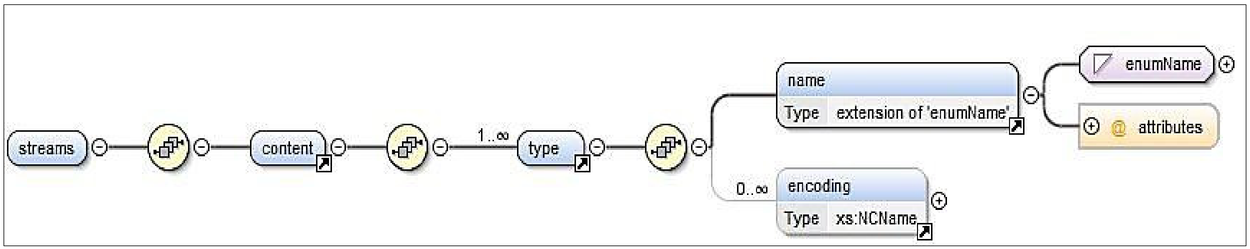
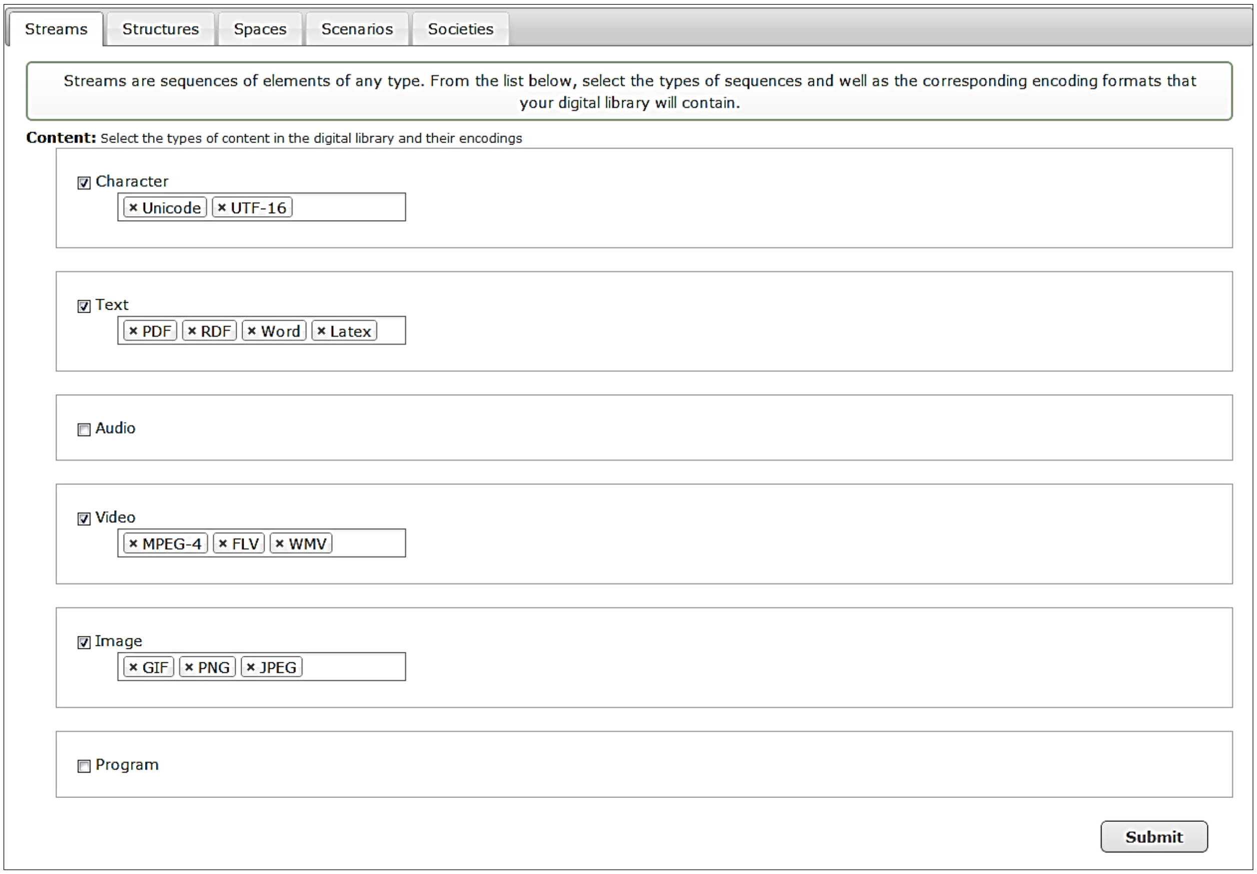


Figure 23. XML schema representation of streams in a DL

The stream model is represented in the user interface of the tool as shown in Figure 24.



Streams are sequences of elements of any type. From the list below, select the types of sequences and well as the corresponding encoding formats that your digital library will contain.

Content: Select the types of content in the digital library and their encodings

☑ Character
× Unicode × UTF-16

☑ Text
× PDF × RDF × Word × Latex

☐ Audio

☑ Video
× MPEG-4 × FLV × WMV

☑ Image
× GIF × PNG × JPEG

☐ Program

Figure 24. User-interface for describing the stream model of the digital library

Checkboxes, on selection, open a multi-selection menu for the encoding options corresponding to the sequence selected. Users can select all the types of content the digital library contains. This gives the team members an idea as to what kind of a digital library is being built. For example, a purely image-based digital library will have image collections and some accompanying text whereas a nation-wide digital library has contents of all types and formats.

5.1.2. Describing the Structures Model

The structural model of a digital library specifies the types of collections it contains and the metadata standards followed for those collections. It has information about the type of collection and sequences included in the collection. These sequences are derived from those described by the streams model of the digital library. For example, if the digital library contains images in the JPEG format, then one example of a sequence any collection in this digital library would include is “Image:JPEG”. The 5SL representation of the structures in a digital library is presented in the schema in Figure 25. The complete XML schema is presented in Appendix B.

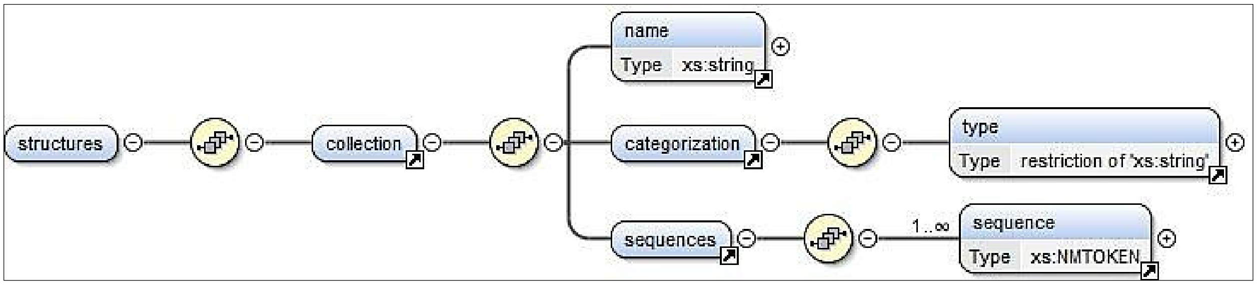
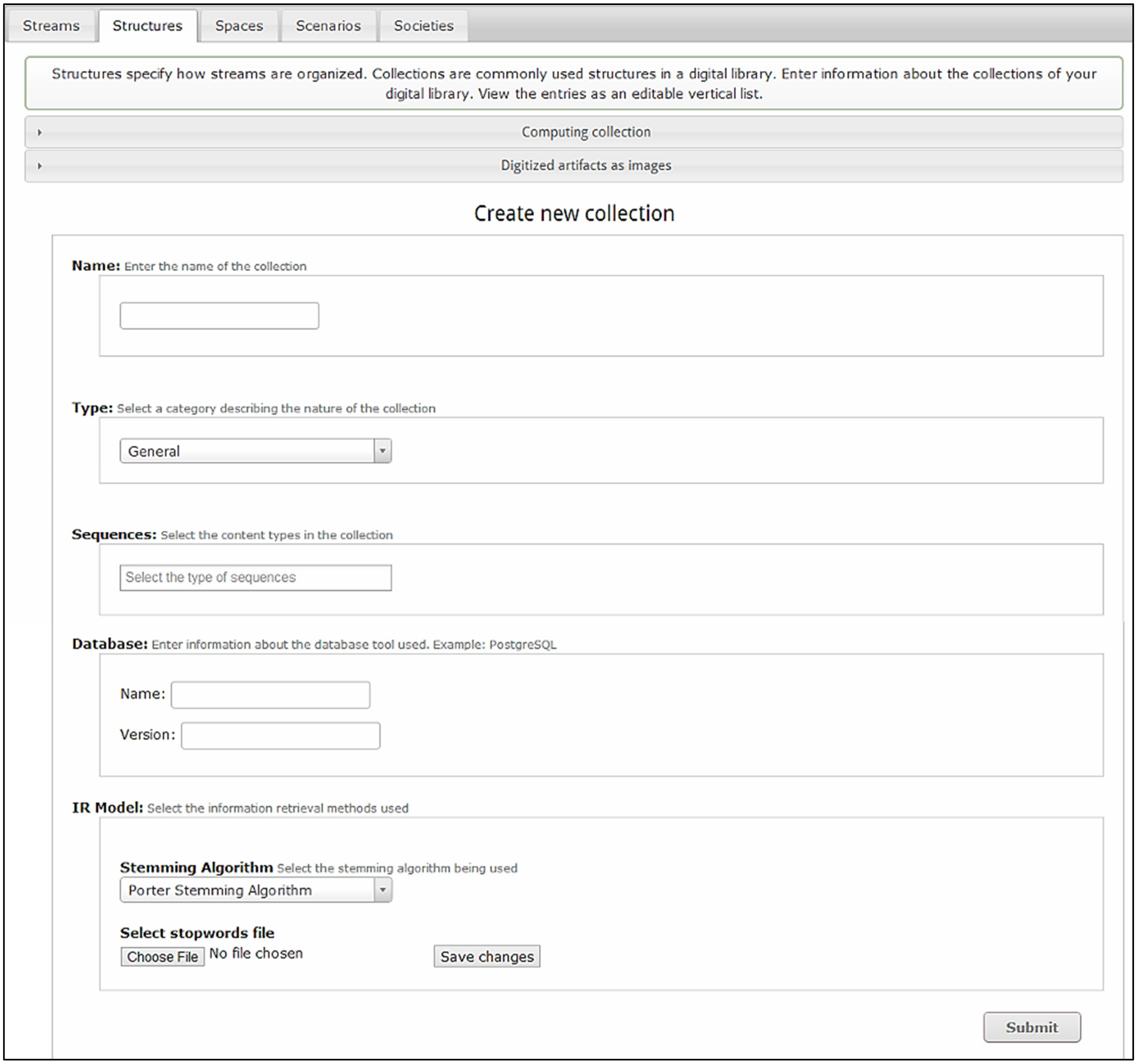


Figure 25. XML schema representation of structures in a DL

Figure 26 presents the user-interface that is used to prompt the user into deriving the structural model of the digital library.



Structures specify how streams are organized. Collections are commonly used structures in a digital library. Enter information about the collections of your digital library. View the entries as an editable vertical list.

Create new collection

Name: Enter the name of the collection

Type: Select a category describing the nature of the collection
General

Sequences: Select the content types in the collection
Select the type of sequences

Database: Enter information about the database tool used. Example: PostgreSQL
Name:
Version:

IR Model: Select the information retrieval methods used

Stemming Algorithm Select the stemming algorithm being used
Porter Stemming Algorithm

Select stopwords file
No file chosen

Figure 26. User-interface for describing the structural model of the digital library

5.1.3. Describing the Spatial Model

The spatial model describes information about the underlying digital library spaces and the operations on them. It includes information about the user-interface space used to present the digital library. The 5SL representation of the spaces in a digital library is presented in the schema in Figure 27. The complete XML schema is presented in Appendix B. Figure 28 illustrates the tool's user interface for describing the spatial model of a digital library.

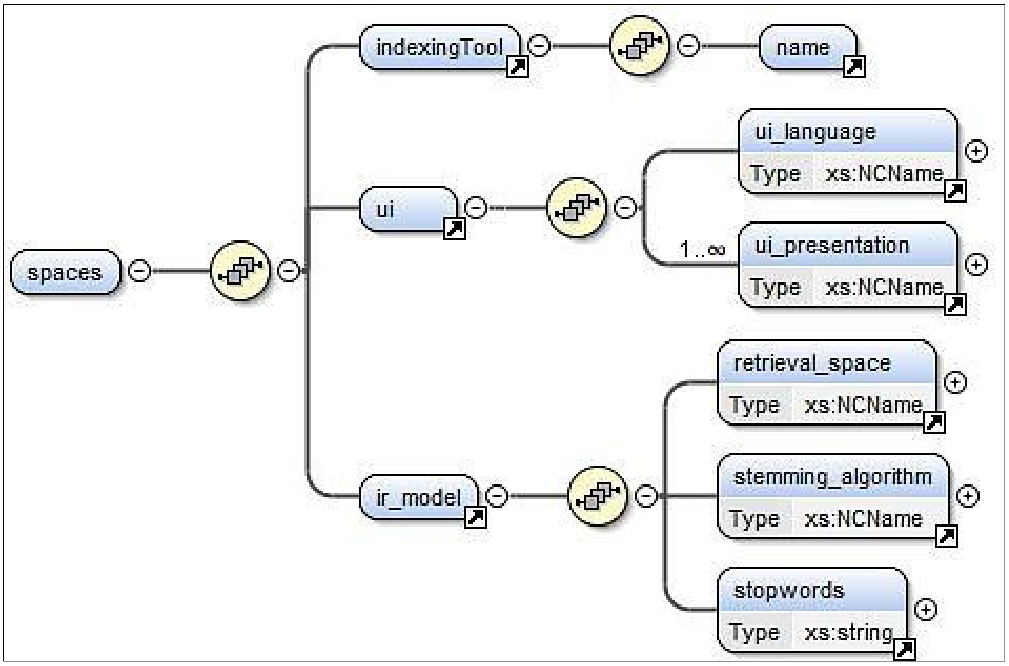
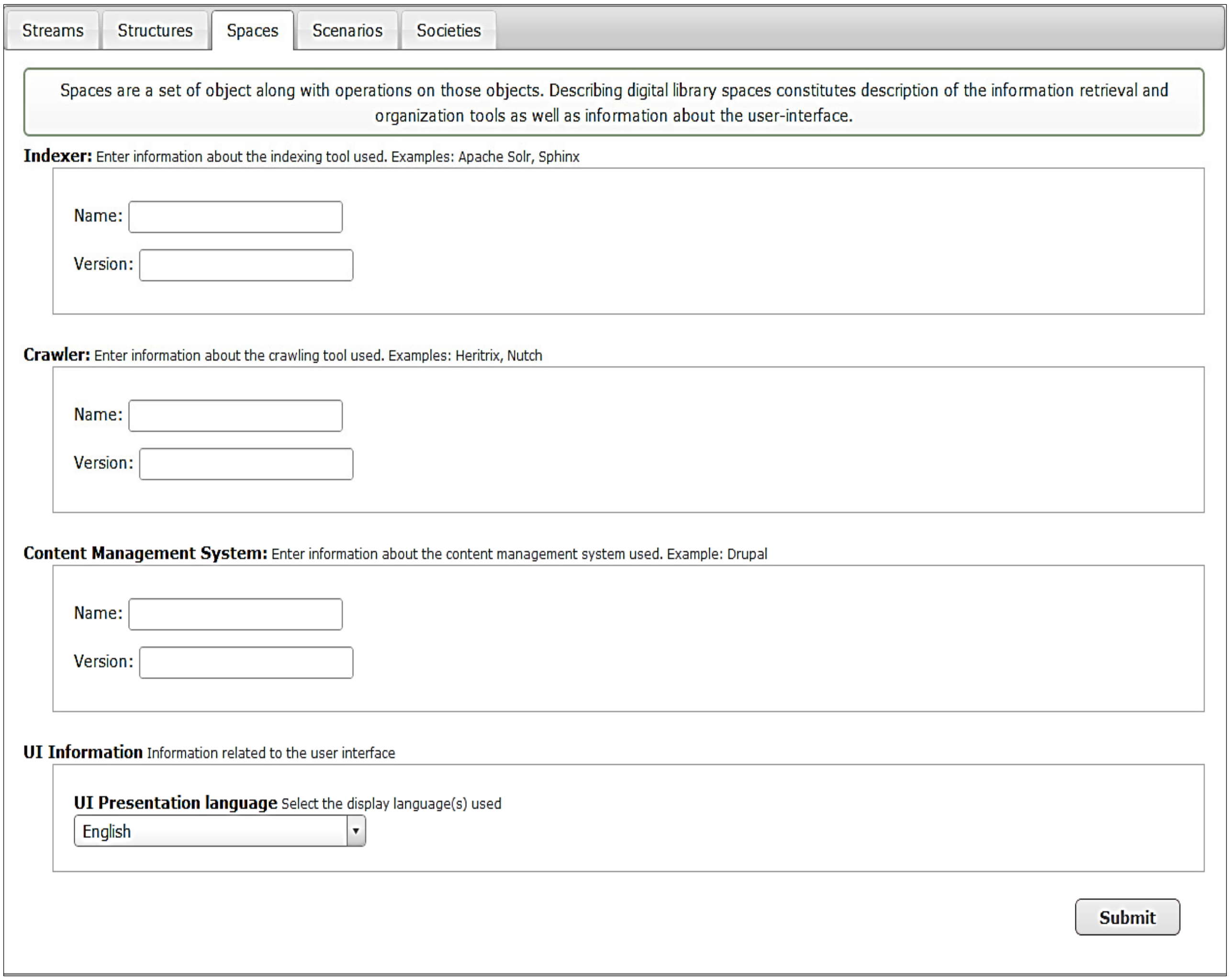


Figure 27. XML schema representation of spaces in a DL

As Figure 28 depicts, information about the software, such as crawler and indexer, to build the digital library, is recorded in the space model. Such information helps sustain the project plan for purposes such as imitating the plan in future projects.



Spaces are a set of object along with operations on those objects. Describing digital library spaces constitutes description of the information retrieval and organization tools as well as information about the user-interface.

Indexer: Enter information about the indexing tool used. Examples: Apache Solr, Sphinx

Name:

Version:

Crawler: Enter information about the crawling tool used. Examples: Heritrix, Nutch

Name:

Version:

Content Management System: Enter information about the content management system used. Example: Drupal

Name:

Version:

UI Information Information related to the user interface

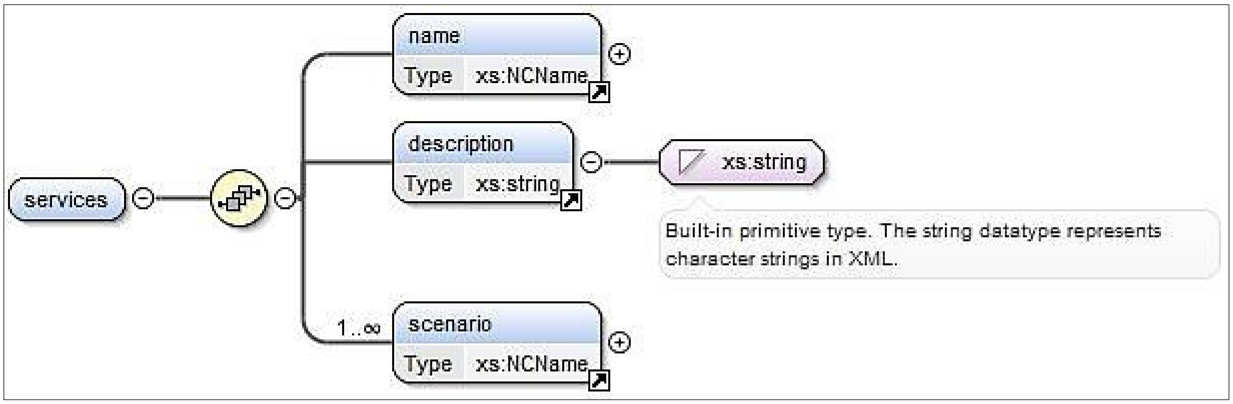
UI Presentation language Select the display language(s) used

English

Figure 28. User-interface for describing the spatial metamodel of the digital library

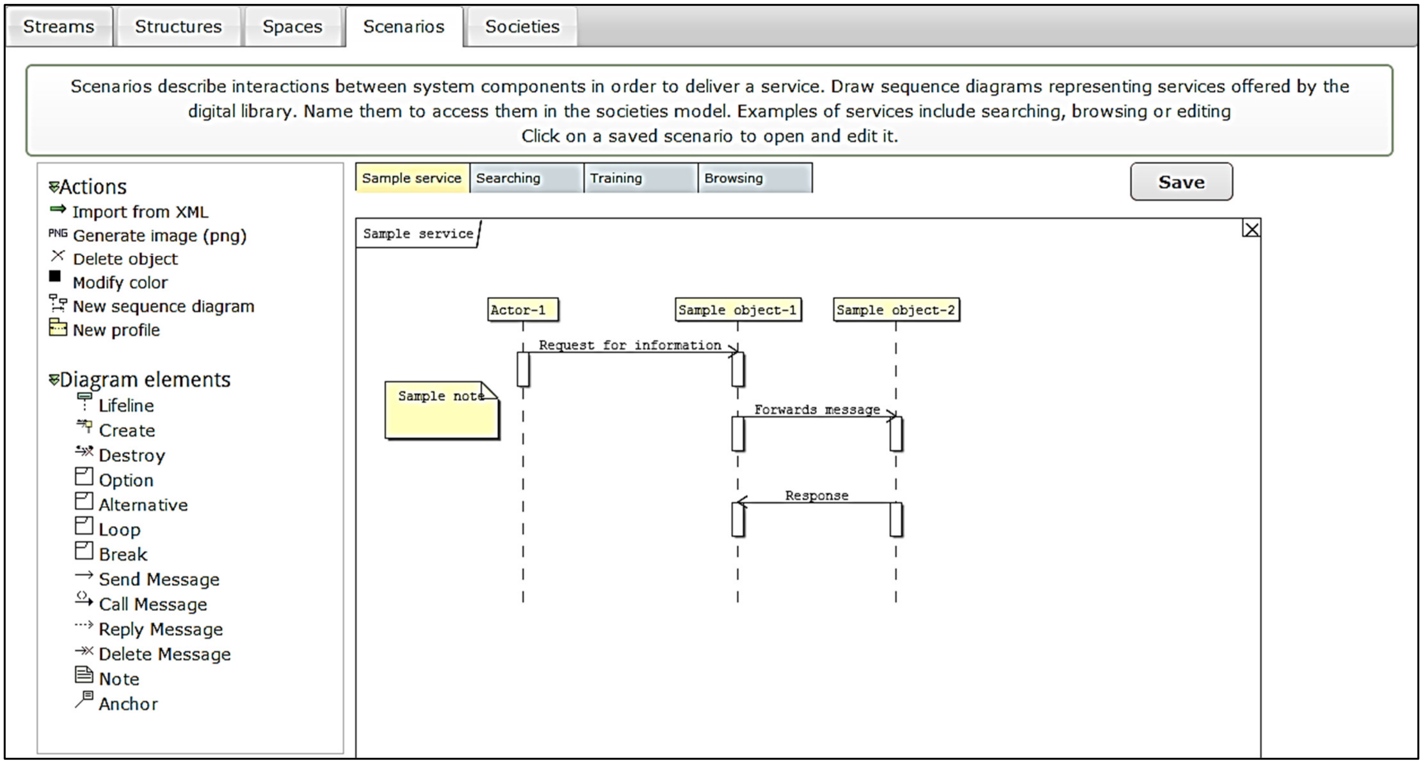
5.1.4. Describing the Scenarios Model

Scenarios tell a story about the services the digital library provides. They are most effectively described using a sequence diagram. A visual UML tool was embedded in the tool so that sequence diagrams can be best represented through illustrations rather than text. Each sequence diagram represents a service that the DL should provide, and the kind of interactions that take place for the same. Any number of such diagrams can be created and saved. The 5SL representation of the scenarios in a digital library is presented in the schema in Figure 29. The complete XML schema is presented in Appendix B. Figure 30 illustrates the tool's user interface for describing the scenarios model of a digital library.



Built-in primitive type. The string datatype represents character strings in XML.

Figure 29. XML schema representation of scenarios in a DL



Scenarios describe interactions between system components in order to deliver a service. Draw sequence diagrams representing services offered by the digital library. Name them to access them in the societies model. Examples of services include searching, browsing or editing
Click on a saved scenario to open and edit it.

Figure 30. User-interface for describing the scenarios model of the digital library

5.1.5. Describing the Societal Model

The societal model of a digital library specifies the users involved in interaction with it (i.e., the communities involved in the DL). They are categorized as groups based on their roles in the digital library. These groups can participate in any number of services described by the scenario model. Figure 32 shows the user-interface for capturing the societal model of the digital library. Each user group can maintain a record of their goals for the project, and can identify the interactions (scenarios) of which they are a part. The 5SL representation of the societies in a digital library is presented in the schema in Figure 31. The complete XML schema is presented in Appendix B.

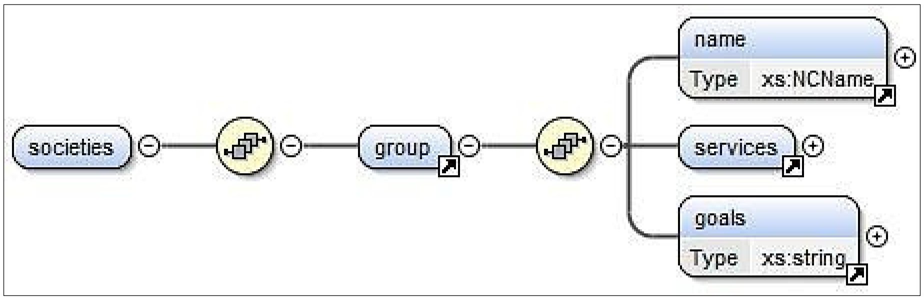
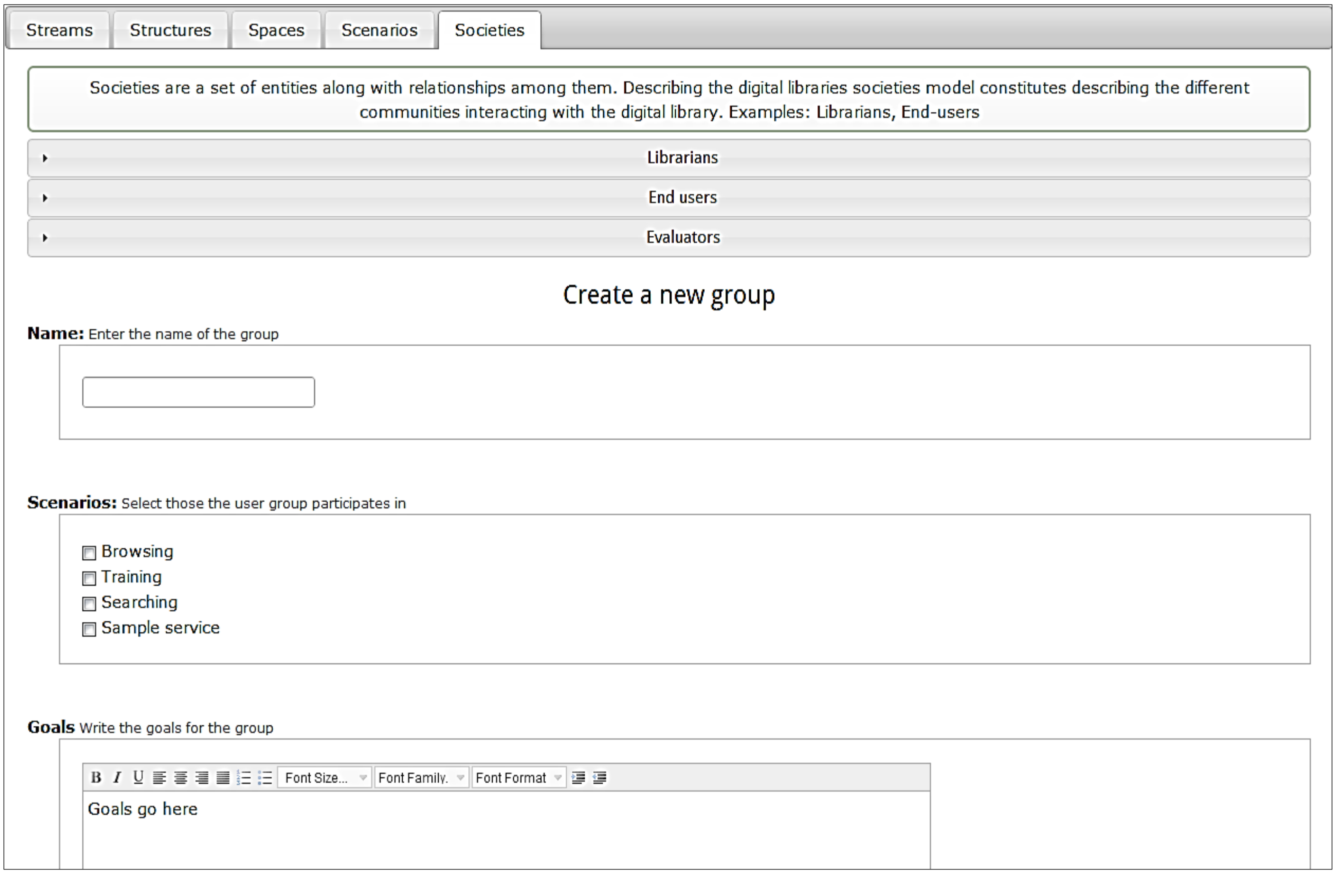


Figure 31. XML schema representation of societies in a DL



Societies are a set of entities along with relationships among them. Describing the digital libraries societies model constitutes describing the different communities interacting with the digital library. Examples: Librarians, End-users

Create a new group

Name: Enter the name of the group

Scenarios: Select those the user group participates in

□ Browsing
□ Training
□ Searching
□ Sample service

Goals Write the goals for the group

Goals go here

Figure 32. User-interface for describing the societal model of the digital library

5.2. 5SL Extension to Describe Project Management

This 5SL extension enables specification of a digital library planning system in five dimensions:

1) Information (text, audio, video, program or image) that contributes to the DL planning (project proposal documents)—Streams model
2) Structuring and grouping of this information—Structural model
3) Presentations of this information so as to allow for operations on it—Spatial model
4) Sequence of events in the planning of a digital library—Scenario model
5) People and groups involved in planning a digital library—Societal model

Planning a digital library is a complex process. It involves specification of the digital library's requirements, how the work is structured, how this information can be accessed easily, the interactions between the different elements of planning, and how people interact when building the digital library.

To explain the nature and challenges of the project management aspects of a digital library, 5S was used as the modeling framework for abstracting and generalizing the digital library planning processes. Table 8 is the result of using 5S to describe the project management entities. Column 2 specifies the several management related constructs that are present in the real world and Column 3 lists the corresponding representations in the tool.

Table 8. 5S modeling of a project management system

5S constituents	**Real-world representations**	**Representations in the tool**
Streams	Task lists, Project proposals, Contracts, Planning documents	REST objects Tasks Resources
Structures	Project phases XML documents	Tree-like organization Hierarchical positioning
Spaces	Physical space Conceptual spaces	Tool space: search, sort, create, update and delete
Scenarios	Collaboration between teams on tasks	Links users with tasks Links projects with phases, events
Societies	Project team members	User objects with roles and responsibilities

5.2.1. Streams in a Project Management System

Streams are essentially sequences of information. From a management perspective, streams constitute entities like task lists, proposals and contract documents. The tool casts these static entities into dynamic ones that can be created, updated, searched, sorted, edited and deleted. Figure 33 depicts the 5SL extension schema for describing the streams in a project management system.

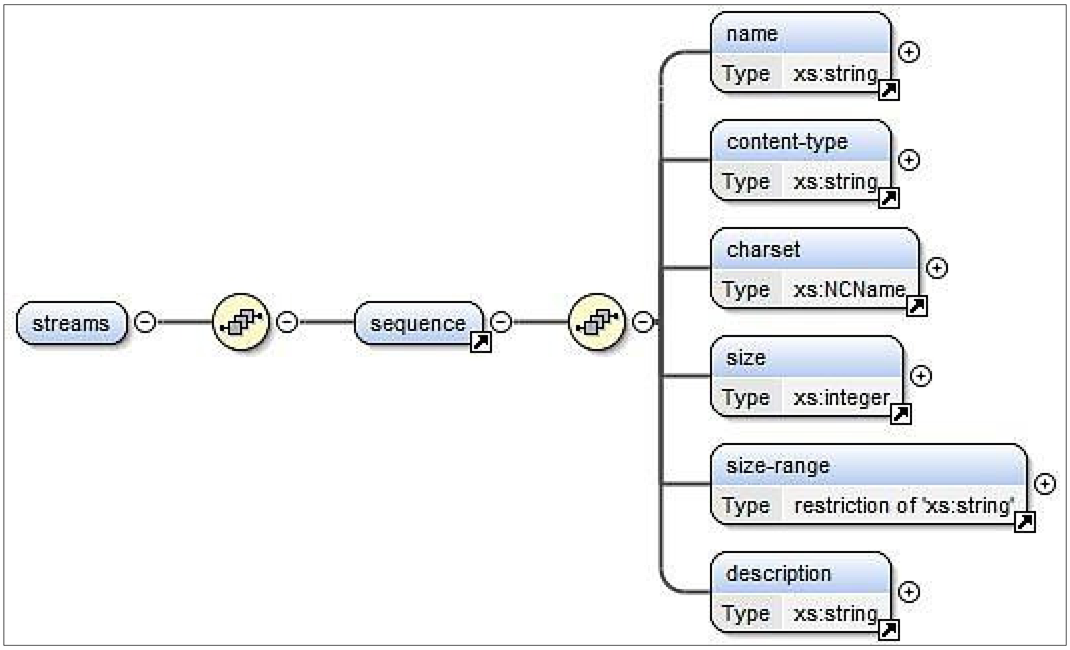


Figure 33. Display of XML schema for streams in a project management system

5.2.2. Structures in a Project Management System

Information streams, often disconnected in the physical world, are represented in relation to other elements with the use of structures. XML representations of tasks, and hierarchical organizations of digital library phases, are some examples of how the structural model in the tool can be described. Figure 34 depicts the 5SL extension schema for describing the structures in a project management system. The phases involved in building a digital library can be structured hierarchically to include other sub-phases.

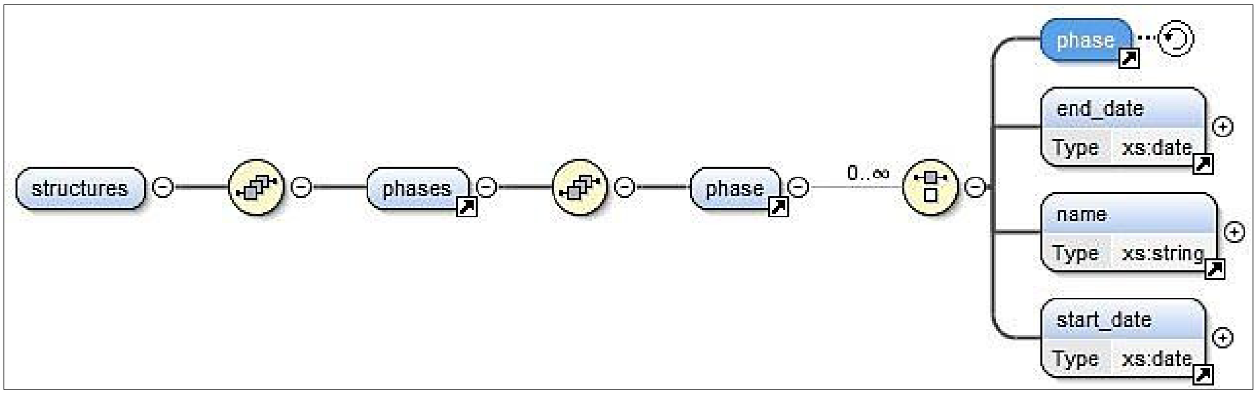


Figure 34. Display of XML schema for structures in a project management system

5.2.3. Spaces in a Project Management System

The management components exist within the tool-space which allows operations like searching, sorting and multi-formatted viewing of data. These operations are performed with the help of the user-interface of the tool as embedded in the user-interface of a Web-browser which in turn allows operations like increasing and decreasing font-size, and copy-paste functions. Figure 35 depicts the 5SL extension schema for describing the spaces in a project management system. A space is defined by the operations that can be performed in it. The "op" element represents the operations that the user-interface of the planning system allows.

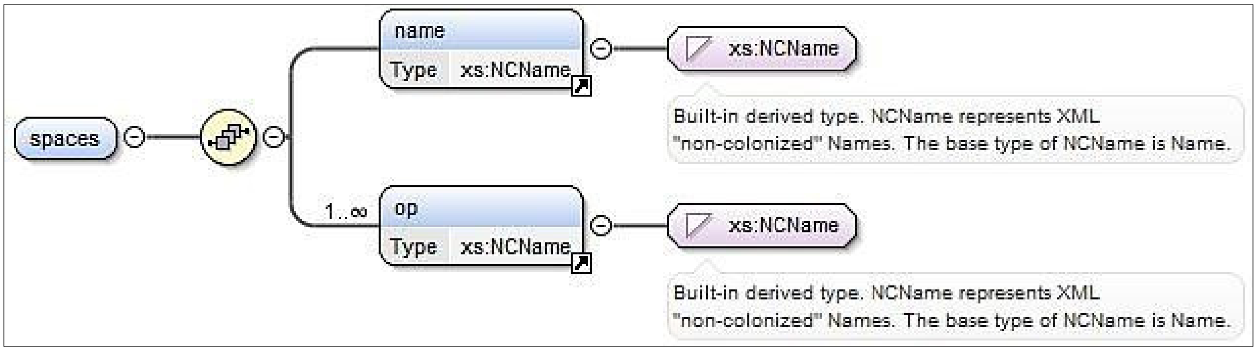

spaces
name
Type xs:NCName
xs:NCName
Built-in derived type. NCName represents XML "non-colonized" Names. The base type of NCName is Name.
1..∞
op
Type xs:NCName
xs:NCName
Built-in derived type. NCName represents XML "non-colonized" Names. The base type of NCName is Name.

Figure 35. Display of XML schema for spaces in a project management system

5.2.4. Scenarios in a Project Management System

Collaboration scenarios between the team members arise especially during the planning stage of a digital library. These scenarios are captured in the form of database relation joins between different object models. For example, a user is assigned the task of building a supplementary collection. This association is encoded in the tool as an assignment relationship between the "User" and the "Task" models. Figure 36 depicts the 5SL extension schema for describing the scenarios in a project management system. Assignment of tasks to team members building the DL constitutes possible scenarios in the planning system. Scenarios also lay out the rules by which the system should work. Digital library policies are examples of such standards that should be highlighted in the planning phase.

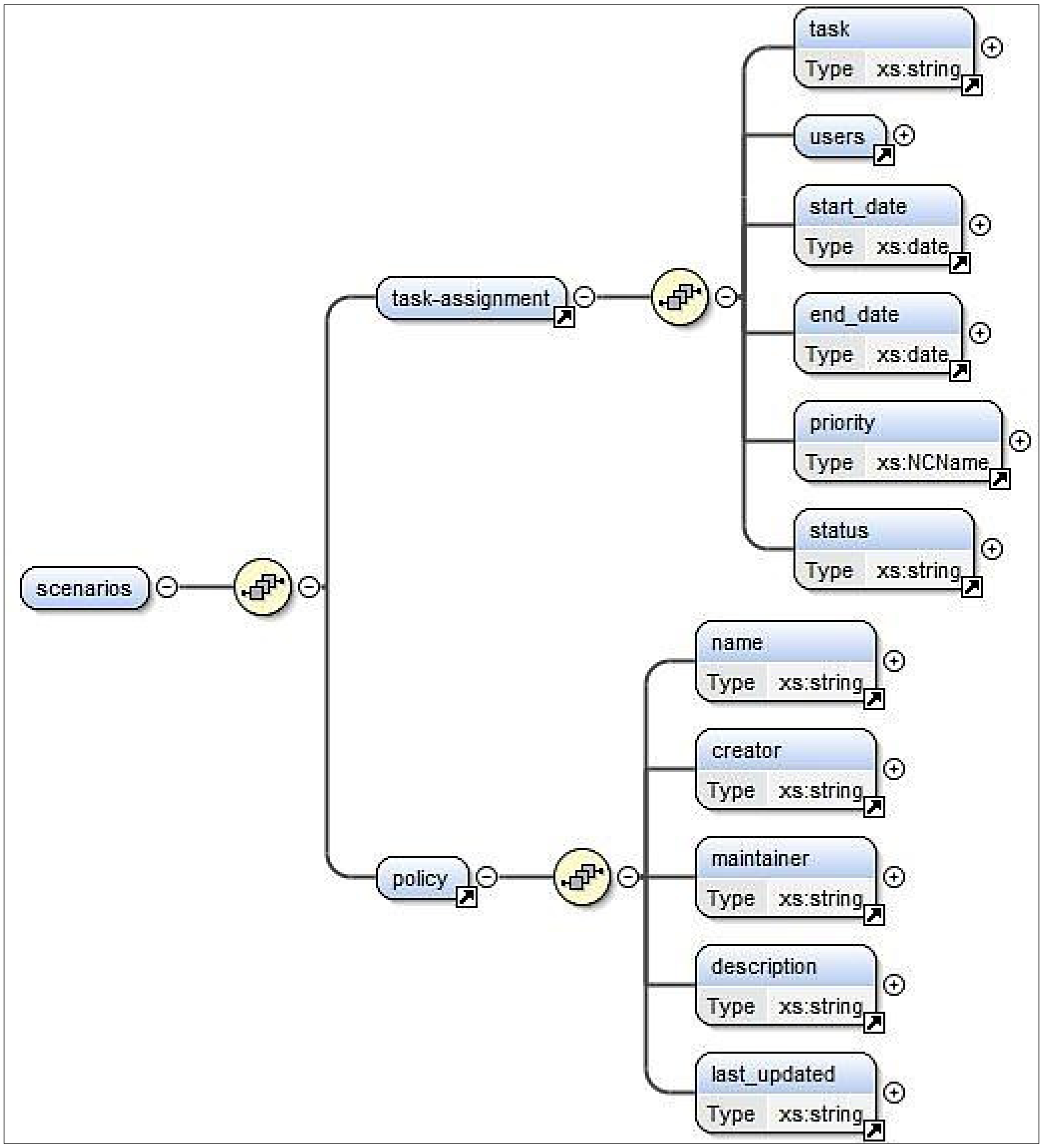


Figure 36. Display of XML schema for scenarios in a project management system

5.2.5. Societies in a Project Management System

The digital library project team involved in the planning, implementation, deployment and evaluation stages constitutes the societal model. This team is represented as the "user" model in the tool, and it exists as the audience for the tool. Societies, apart from people, also constitute hardware and software. The server hosting the tool application and the Web-browser, which allows the interactions to take place with the tool, are also part of the tool's society model. Figure 37 depicts the 5SL extension schema for describing the societies in a project management system. Societies in the planning system include the stakeholders of the system and the team primarily responsible for bringing the digital library to life. The 5SL extension of societies for the project management system represents these people and their attributes.

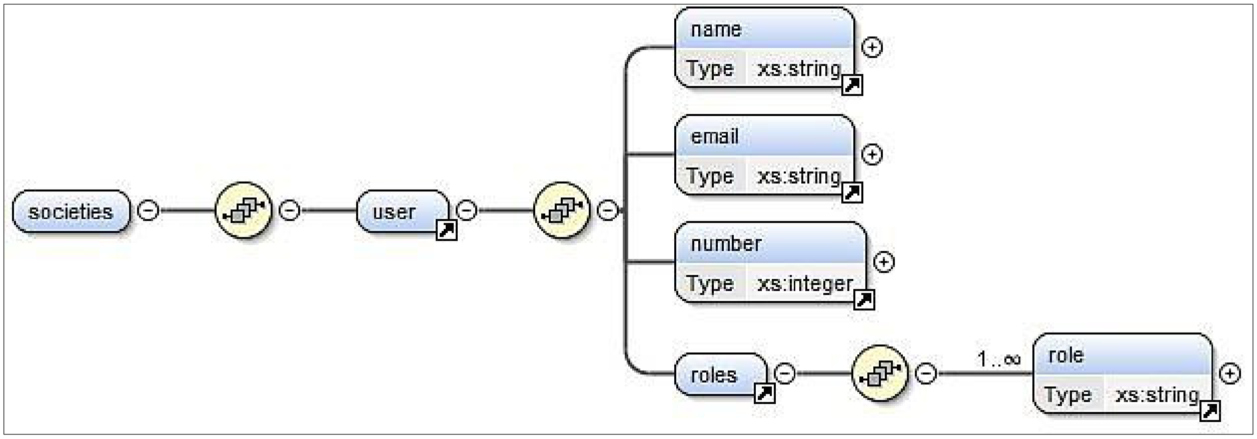


Figure 37. Display of XML schema for societies in a project management system

6. Formative Evaluation of the Digital Library Development Management Tool

The goal of the evaluation study is to test the feasibility for using the tool for real-world digital libraries. The feedback obtained from the study can be used to improve the tool further. The approach was to model the digital libraries' management systems as well as their 5S descriptions. Sections 6.1 and 6.2 describe the formative evaluation case studies and examine the feedback obtained from the study.

6.1. Case study: Crisis, Tragedy and Recovery Network (CTRnet) Digital Library

6.1.1. Introduction to CTRnet

The CTRnet project [14] was undertaken at Virginia Tech with the aim of integrating content related to natural or man-made disasters for affected communities (victims), information seekers and recovery efforts. Communities involved included emergency response workers, law enforcement agencies, government officials and community activists. Collections are built by crawling the Internet and social media sites like Twitter to collect any information about a particular disaster using seed keywords. Collections are stored and made accessible to users. A considerable amount of information has been collected and the current stage of the project is building and implementing the digital library to allow users to effectively access the information.

Ongoing work includes developing the digital library prototype using Drupal. Implementing a set of services, from browsing and visualizing to backend work like data-extraction and retrieval, will use the LucidWorks Enterprise platform.

Though the CTRnet project officially ended in the summer of 2013, when NSF funding ran out, another NSF-funded project, called IDEAL, began 1 September 2013, and is continuing the CTRnet project in an expanded fashion. Accordingly, for simplicity, only the name CTRnet will be used in this thesis.

About 15–20 people are currently working on the project in various capacities. There are also collaborators from other countries. Current work revolves around getting the digital library running and in use by many people. Communication channels for the project involve media like email and website announcements.

The digital library development management tool is useful in enhancing collaboration and allowing project team members to learn about the progress of ongoing work and completion time estimates on complex or ambiguous tasks.

6.1.2. Approach

An active member of the team was consulted regarding documentation and information about the history, current and future directions of the project. The various nuances of project management in the digital library were identified, and discussed at length. The digital library's project management realm was modelled into the tool by recording information about the users and tasks in the project. The streams, structures, spaces, scenarios and societies of the project were described using the tool.

Feedback about the tool was gathered and many enhancements were suggested to make the tool more versatile for use with real-world digital libraries. The section below is a presentation of the results from developing the management plan for the project, and includes a discussion about possible enhancements.

6.1.3. CTRnet Streams

The crawler and the data extractor building the digital library archive of crisis events collect all types of media from text files to multimedia files like audio and video. The sequences and streams collected include almost all existing types of encoding formats. The tool contained the content types and encoding formats as listed in Table 9. It included only some widely used formats. A suggestion was to include a list of all the internet media types [47] and accordingly this feature was added to the tool. Further, in order to improve the user experience around selecting from a larger number of options, the multi-select drop down was replaced with checkbox-based selection.

Table 9. Previously available encoding options

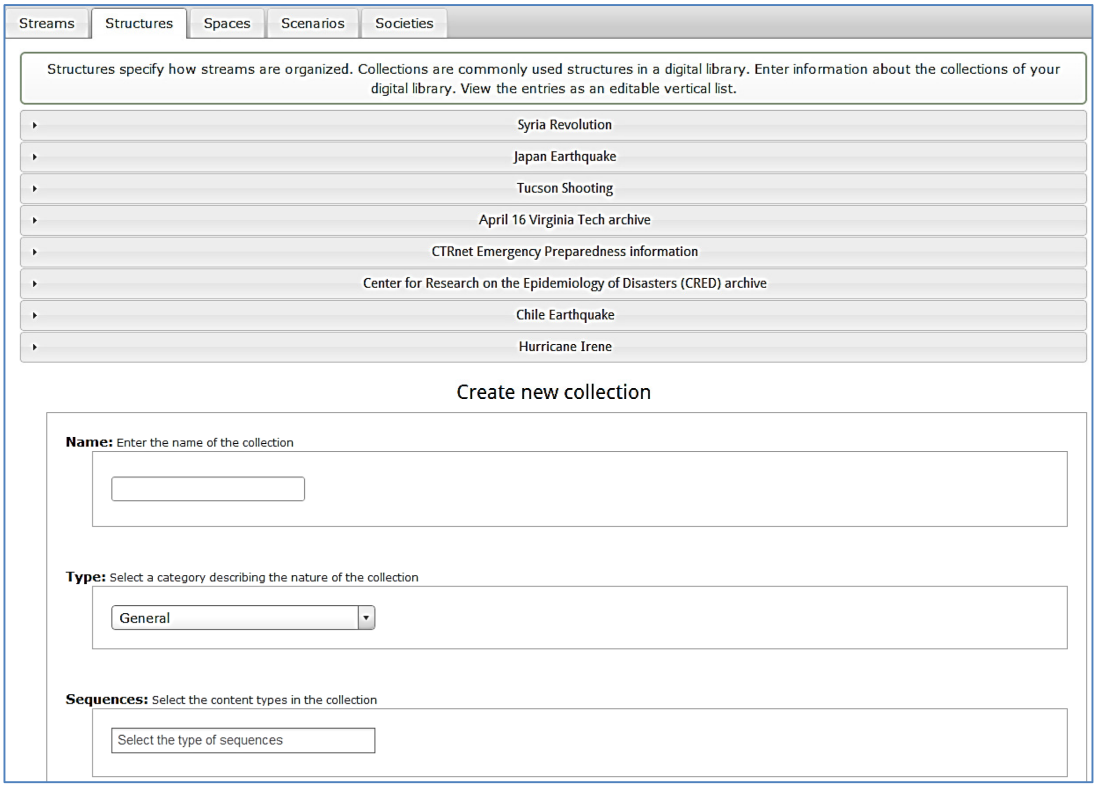
Media Types	**Encoding Options**
Text	Plain text, PDF, RDF, Word, Latex
Audio	MPEG-2, MPEG-4, QuickTime, WAV, MP3, WMA
Video	MPEG-4, AVI, FLV, WMV, MOV
Image	GIF, PNG, JPEG

Table 10. Current comprehensive list of encoding options

Media Types	**Encoding Options**
Text	Cmd, CSS, HTML, Javascript, Plain text, vcard, XML
Audio	Basic, MPEG-2, MPEG-4, QuickTime, WAV, MP3, WMA, Worbis, RealAudio, Linear PCM
Video	MPEG-1, QuickTime, WebM, Ogg, MPEG-4, AVI, FLV, WMV, MOV
Image	GIF, PNG, JPEG, PJPEG, TIFF
Program/ Application	ECMAScript, EDIFACT, Json, Octet, Zip, XHTML

6.1.4. CTRnet Structures

The structures describe the collections in the CTRnet digital library. Figure 38 shows a screenshot of how information about some of the available collections is stored. The feedback from the evaluation was that correlating the management entities like tasks and users with the 5S description is a great way to improve the interoperability and flexibility of the entities. This work is planned to be added to the next version of the tool. Features will include the ability to assign users to a collection and linking tasks that constitute or are related to building a collection to the structural description of the project. Figure 39 depicts the proposed user interface to make this linking possible.



Structures specify how streams are organized. Collections are commonly used structures in a digital library. Enter information about the collections of your digital library. View the entries as an editable vertical list.

- Syria Revolution
- Japan Earthquake
- Tucson Shooting
- April 16 Virginia Tech archive
- CTRnet Emergency Preparedness information
- Center for Research on the Epidemiology of Disasters (CRED) archive
- Chile Earthquake
- Hurricane Irene

Create new collection

Name: Enter the name of the collection

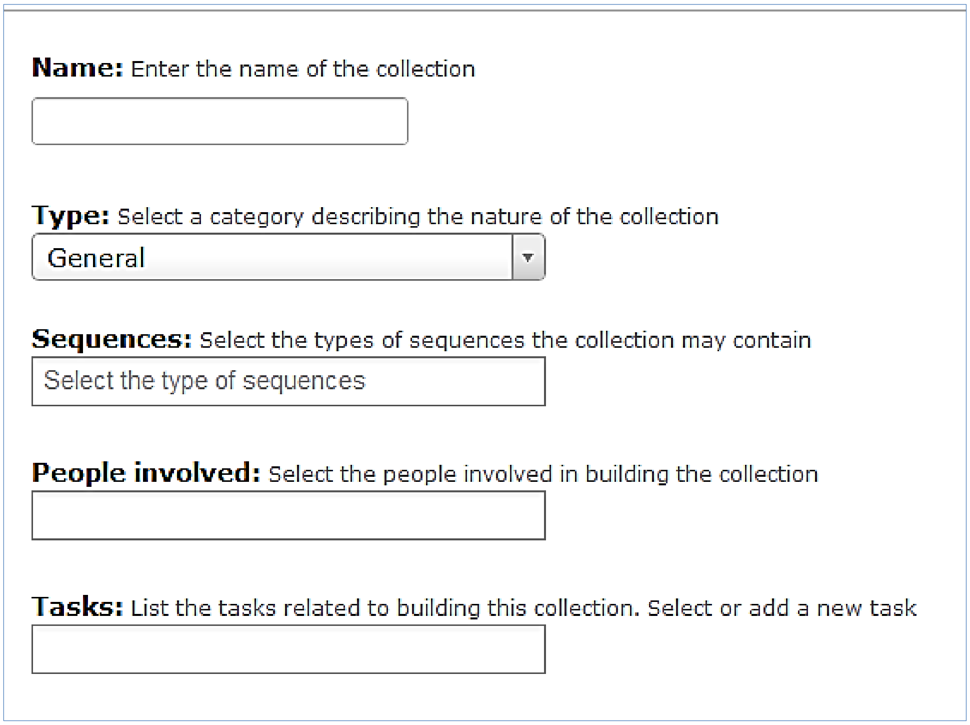
Type: Select a category describing the nature of the collection

General

Sequences: Select the content types in the collection

Select the type of sequences

Figure 38. CTRnet structures



Name: Enter the name of the collection

Type: Select a category describing the nature of the collection

General

Sequences: Select the types of sequences the collection may contain

Select the type of sequences

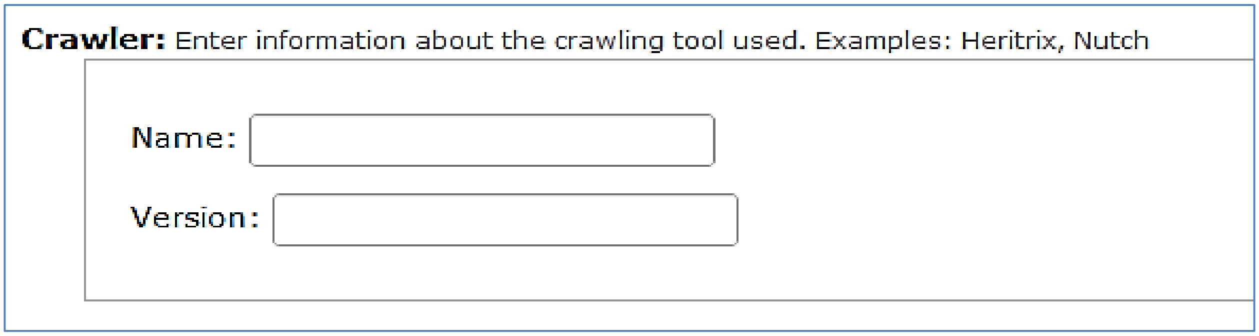
People involved: Select the people involved in building the collection

Tasks: List the tasks related to building this collection. Select or add a new task

Figure 39. Improved model of structures

6.1.5. CTRnet Spaces

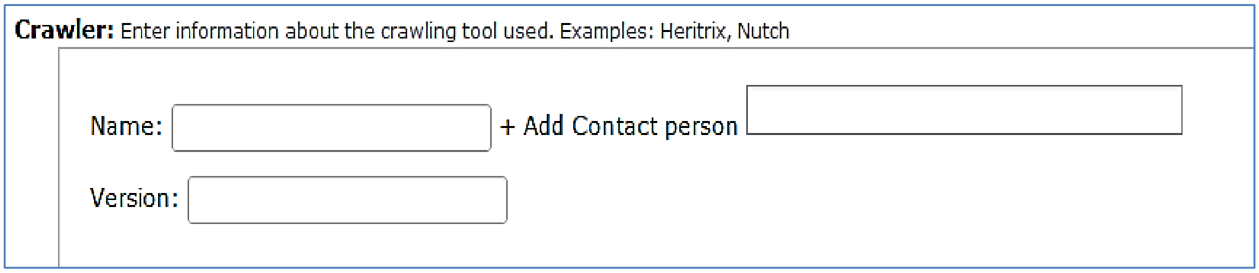
The digital library is embedded in a description of the context of the information organization tools, and of the other software used in building it. The user can enter information about the crawling software and indexer software used as shown in Figure 40.



Crawler: Enter information about the crawling tool used. Examples: Heritrix, Nutch

Figure 40. Current space model

Describing the tool itself may be insufficient when the project team is changing frequently. It is important for every member to record what kind of working knowledge they have to make it easier for people joining the project at a later date. Thus, adding information about the users with knowledge about these tools, as shown in Figure 41, is helpful for identifying who can troubleshoot which software. It is a method for sustaining the project processes and will help in future projects.

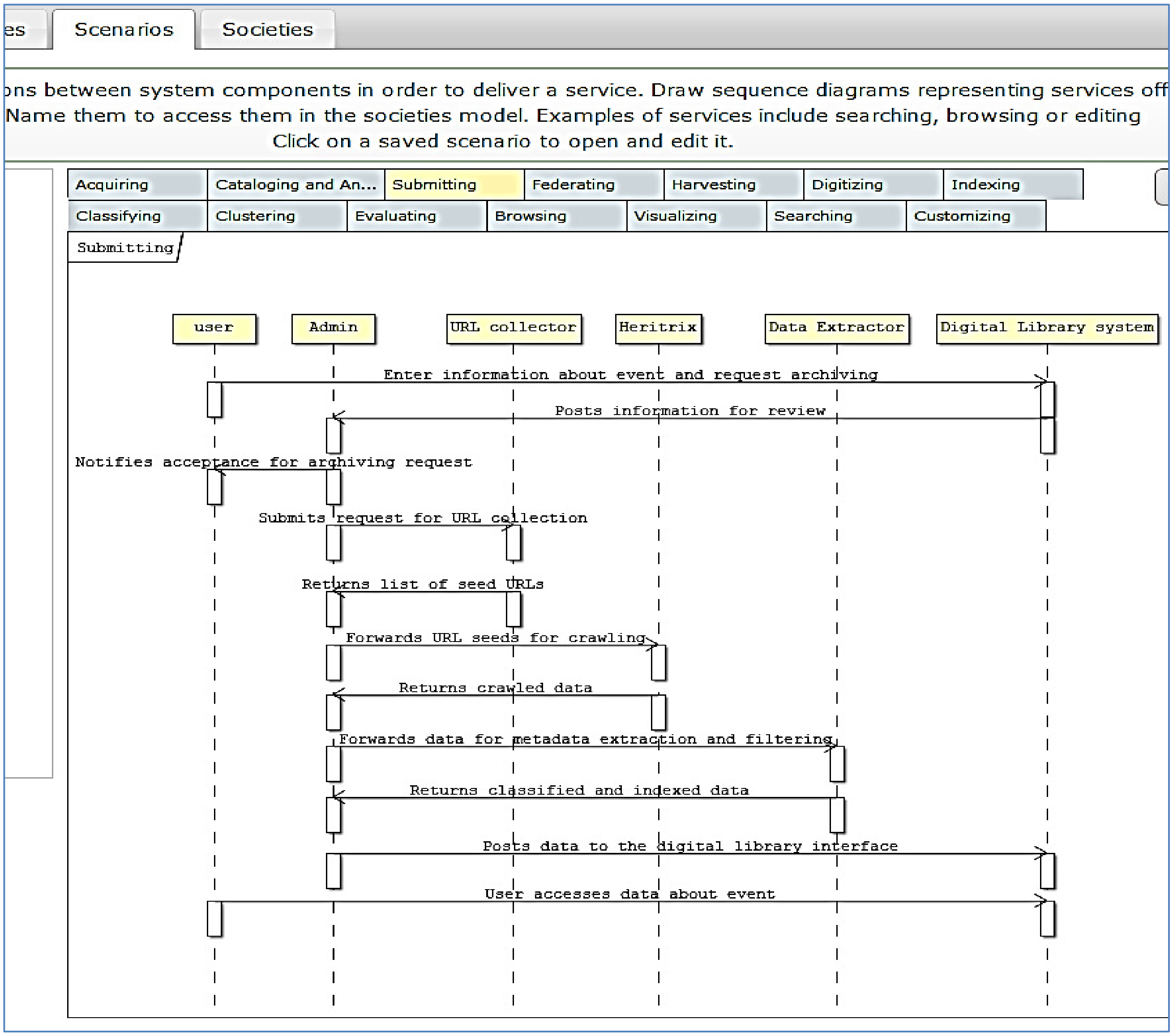


Crawler: Enter information about the crawling tool used. Examples: Heritrix, Nutch

Figure 41. Improved space model

6.1.6. CTRnet Scenarios

There are many services provided by a large digital library like CTRnet. The figure below shows some of the services that represent scenarios in the CTRnet digital library. The sequence diagram shown in Figure 42 depicts the sequence of actions involved in the processing of user-submitted requests for archiving events.



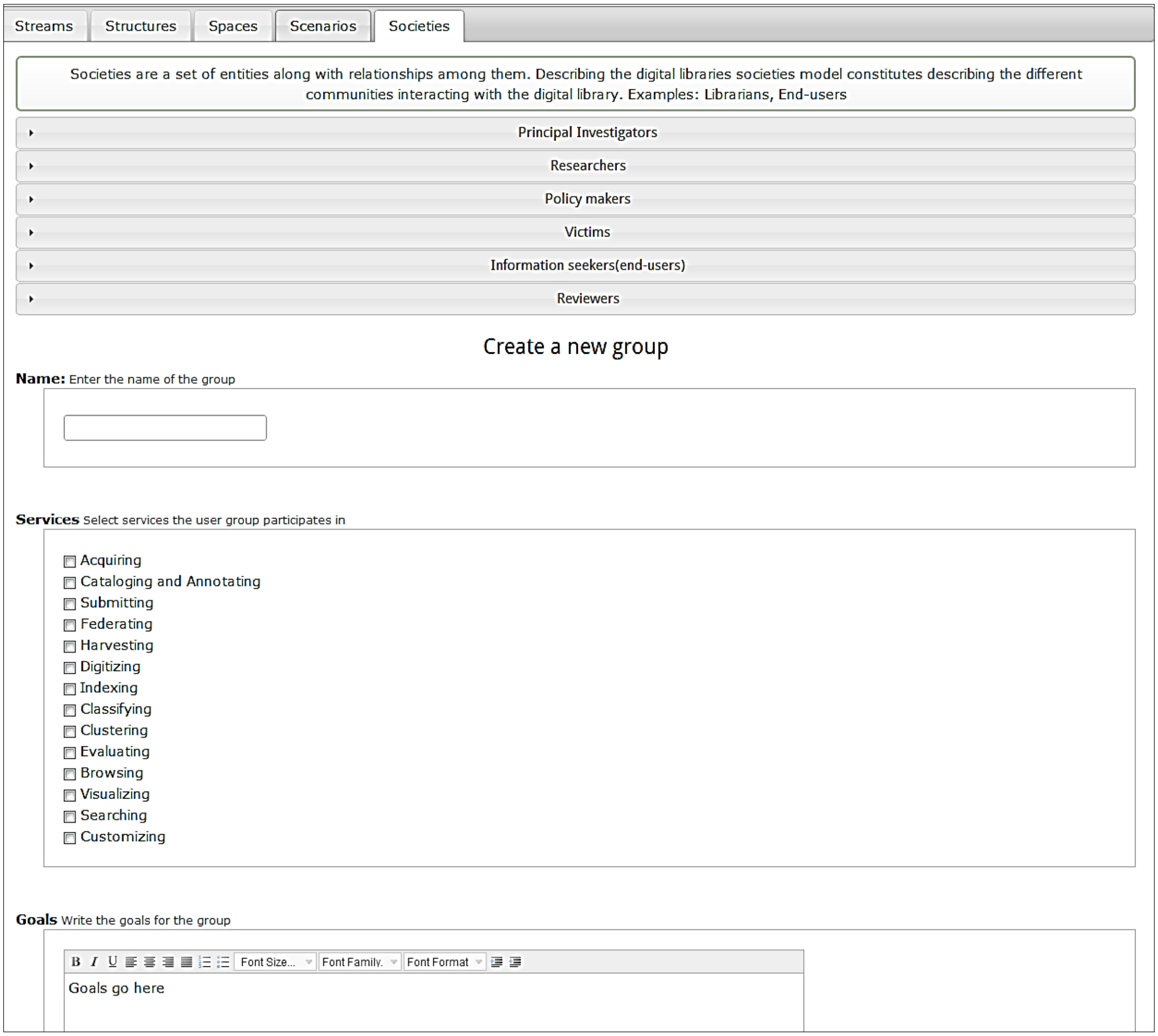
ons between system components in order to deliver a service. Draw sequence diagrams representing services off
Name them to access them in the societies model. Examples of services include searching, browsing or editing
Click on a saved scenario to open and edit it.

es
Scenarios
Societies
Acquiring
Cataloging and An...
Submitting
Federating
Harvesting
Digitizing
Indexing
Classifying
Clustering
Evaluating
Browsing
Visualizing
Searching
Customizing
Submitting
user
Admin
URL collector
Heritrix
Data Extractor
Digital Library system
Enter information about event and request archiving
Posts information for review
Notifies acceptance for archiving request
Submits request for URL collection
Returns list of seed URLs
Forwards URL seeds for crawling
Returns crawled data
Forwards data for metadata extraction and filtering
Returns classified and indexed data
Posts data to the digital library interface
User accesses data about event

Figure 42. Scenario description: User-submitted requests to archive events

6.1.7. CTRnet Societies

The different user groups (communities) interacting with the digital library system constitute the societies model of the CTRnet project. As shown in Figure 43, each user group can participate in any of the scenarios as recorded in the scenario section of the digital library. User groups in the CTRnet project include not only members of the project team, but also people who provide information for building the collections via tweets, news articles and blog posts. First-hand accounts from disaster victims also lead to invaluable collections.



Societies are a set of entities along with relationships among them. Describing the digital libraries societies model constitutes describing the different communities interacting with the digital library. Examples: Librarians, End-users

- Principal Investigators
- Researchers
- Policy makers
- Victims
- Information seekers(end-users)
- Reviewers

Create a new group

Name: Enter the name of the group

Services Select services the user group participates in

□ Acquiring
□ Cataloging and Annotating
□ Submitting
□ Federating
□ Harvesting
□ Digitizing
□ Indexing
□ Classifying
□ Clustering
□ Evaluating
□ Browsing
□ Visualizing
□ Searching
□ Customizing

Goals Write the goals for the group

Goals go here

Figure 43. CTRnet Societies

6.1.8. Tasks

Digital library project tasks include work that is ongoing or on to-do lists. However, there is currently no support for nesting of tasks using a parent-child relationship to divide large tasks into smaller ones. Possibly, there should be more levels of nesting. For example, a 2012 event in the CTRnet project was a Webinar on emergency informatics and digital libraries. Preparing for the webinar was a major undertaking, and there were a number of smaller tasks, which, when allocated to appropriate users, made it a success. The tool therefore was enhanced to support this hierarchical structuring of tasks.

6.1.9. Resources

Currently users can upload resources to projects, but not to specific tasks or events. A resource like a project's proposal document is relevant in the scope of the entire project, whereas a technical paper on collection development might only be relevant to a particular task about building collections. However, the tool currently places all resources in the scope of the project. The proposed enhancement that will be in the next version of the tool is to provide an option to upload and associate resources such as text files, images and code snippets to entities like tasks, events and phases.

6.2. Case Study: National Digital Library

The second case study for evaluating the tool was a generic digital library-National Digital Library.

6.2.1. Approach

A study of typical processes and phases that constitute digital library development was conducted. A current member of the Digital Library Research Laboratory participated in the formative evaluation. The outcome of the evaluation and the feedback is presented in the sections following. The users, phases and tasks for the digital library were imported in an XML format using the XML schema presented in Section 3.1.4 and Appendix A. Sections 6.2.2 and 6.2.3 present some of the screenshots resulting from developing a management model for the National Digital Library (NDL) as well as the feedback and suggestions for enhancing the tool.

6.2.2. NDL Phases

The digital library development management tool was effective in presenting the phases, and it allowed users to nest phases within other higher-level phases. The phases also can be created using a guiding list of phases that a user can choose from to include in the project. Figure 44 represents the index page of phases. There are different ways of viewing the project's phases, namely, timeline, list, table or hierarchical tree formats as shown in Figure 44.

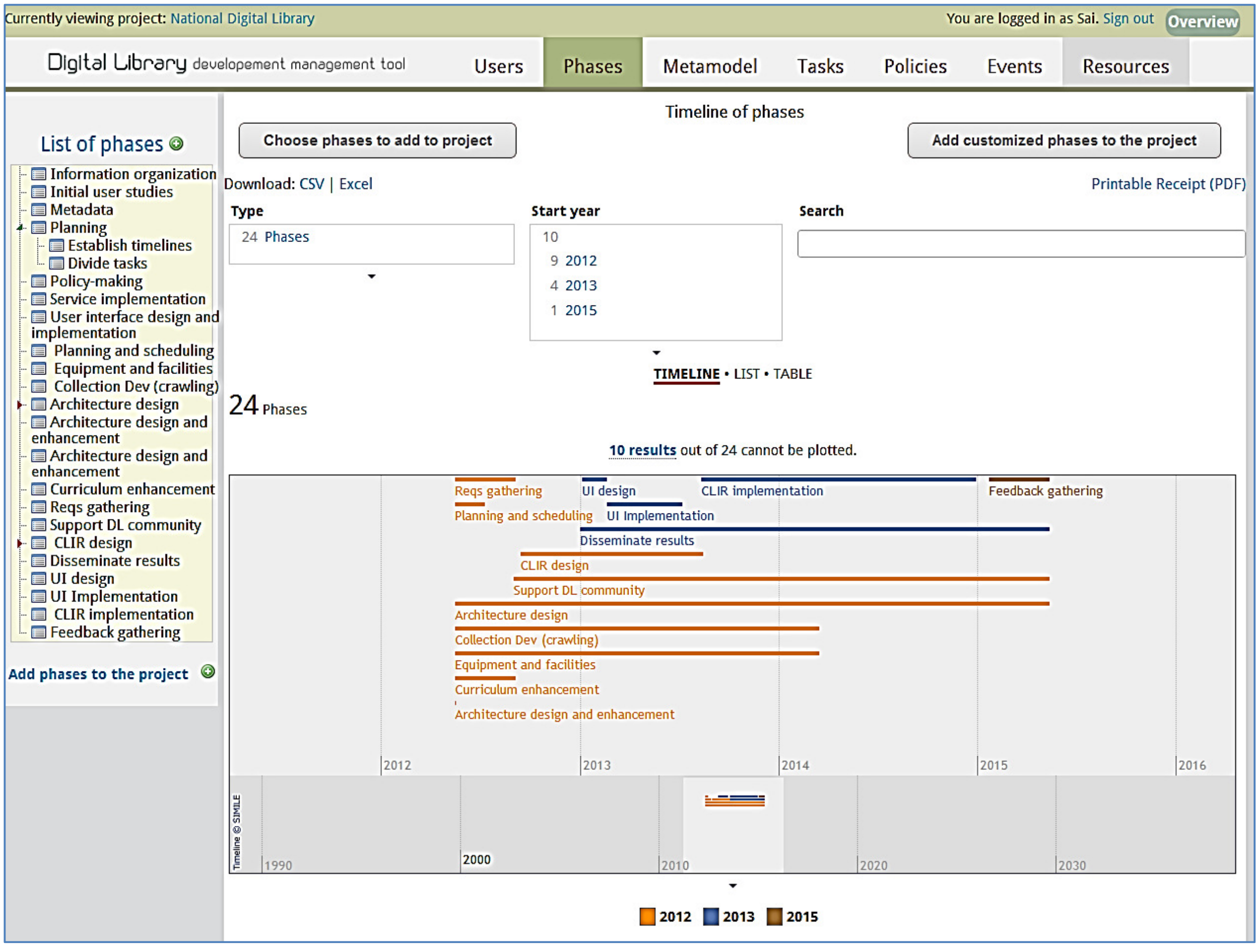
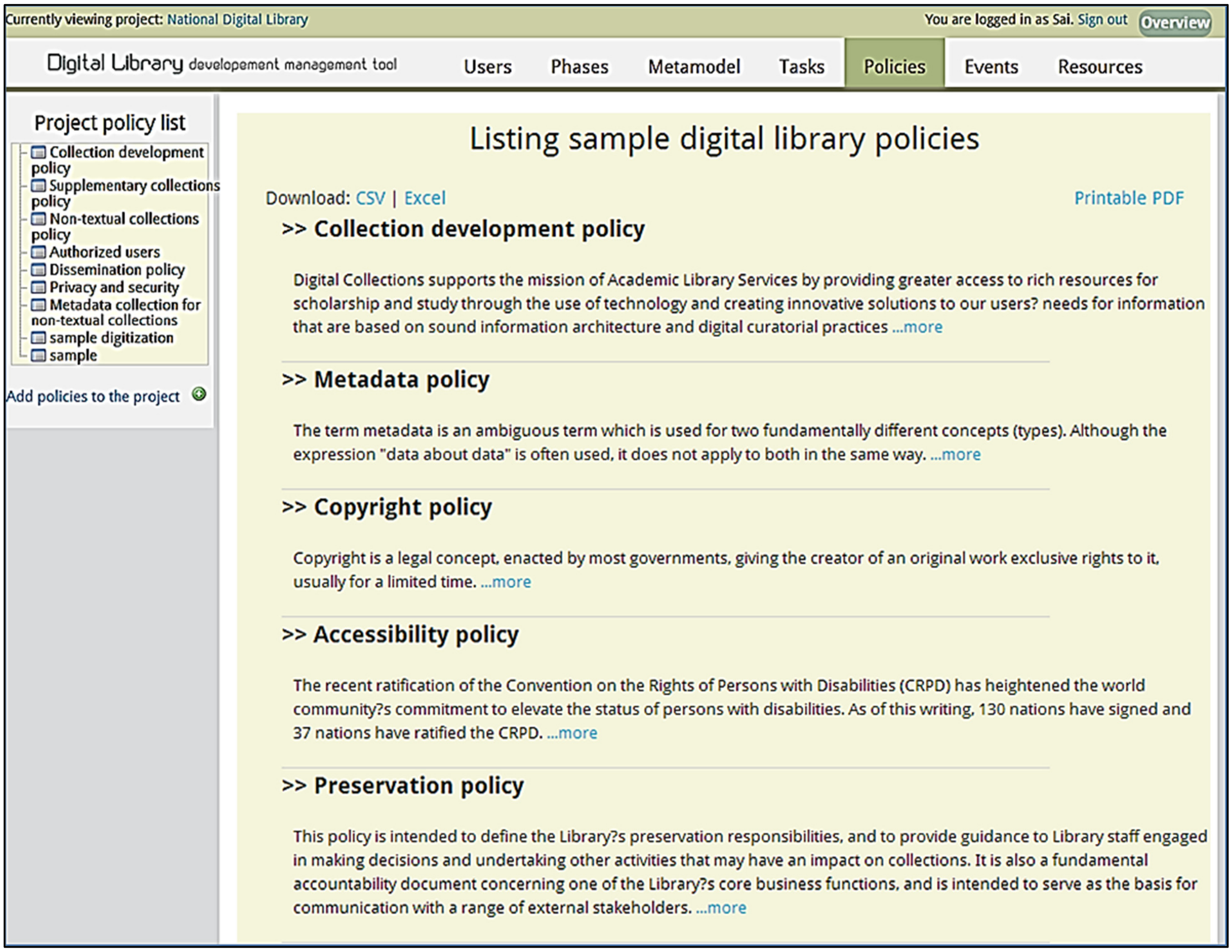


Figure 44. Phases in the NDL project

6.2.3. NDL Policies

Digital library policies define the rules and standards the digital library should adhere to. As project management entities, policies guide the development of a project. Well-documented policies ensure that all the team members are aware of the principles along which the digital library should be developed. Figure 45 is a screenshot of the listing page for policies in the National Digital Library project. The section on the left of the screenshot shows the listing of the project's policies. The section on the right is a list of sample policies that digital libraries typically define for themselves. A project team can utilize this list as a guide around which to formulate policies for their own digital library. Since these samples are used by all users of the tool, new ones can be added or existing ones updated by the maintainer of the tool. The "more" link on a policy will open up the complete description of the policy and provide links to use that description as a template for a new custom policy.


Currently viewing project: National Digital Library
You are logged in as Sai. Sign out
Overview
Digital Library developement management tool
Users
Phases
Metamodel
Tasks
Policies
Events
Resources
Project policy list
Collection development policy
Supplementary collections policy
Non-textual collections policy
Authorized users
Dissemination policy
Privacy and security
Metadata collection for non-textual collections
sample digitization
sample
Add policies to the project
Download: CSV | Excel
Printable PDF

Listing sample digital library policies

>> Collection development policy

Digital Collections supports the mission of Academic Library Services by providing greater access to rich resources for scholarship and study through the use of technology and creating innovative solutions to our users? needs for information that are based on sound information architecture and digital curatorial practices ...more

>> Metadata policy

The term metadata is an ambiguous term which is used for two fundamentally different concepts (types). Although the expression "data about data" is often used, it does not apply to both in the same way. ...more

>> Copyright policy

Copyright is a legal concept, enacted by most governments, giving the creator of an original work exclusive rights to it, usually for a limited time. ...more

>> Accessibility policy

The recent ratification of the Convention on the Rights of Persons with Disabilities (CRPD) has heightened the world community?s commitment to elevate the status of persons with disabilities. As of this writing, 130 nations have signed and 37 nations have ratified the CRPD. ...more

>> Preservation policy

This policy is intended to define the Library?s preservation responsibilities, and to provide guidance to Library staff engaged in making decisions and undertaking other activities that may have an impact on collections. It is also a fundamental accountability document concerning one of the Library?s core business functions, and is intended to serve as the basis for communication with a range of external stakeholders. ...more

Figure 45. Policies in the NDL project

6.2.4. NDL Users

One important issue addressed in the evaluation was that of cold-start – when all of the team members of a digital library project are not registered in this tool, then it becomes difficult for members to form a complete picture of the project's management processes. Currently this is addressed by allowing a user to create other "dummy" users that act as any other user object in the tool. For example, they can be assigned tasks. This approach is, however, not a favored solution because of the confusion that can arise when the users eventually register.

The preferred solution that was suggested during the evaluation was to use an invite-by-email solution. This would work by having a registered user send a registration invitation to the user whose participation is required in the project. Emails also can be used for other kinds of communication such as updates on events or updates on an important task's status.

6.2.5. Access Privileges

As it stands, the tool allows any person in the team to edit or delete any type of information in the project. This was identified as a security concern. For example, if a user leaves a project and this change is not updated in the tool, then there is a chance of a security breach during this time gap. Introducing access privileges is a way of restricting access to the information. A possible way of allocating privileges is shown in Table 11.

Table 11. Proposed breakdown of access privileges

User group	Access permissions
Administrator	Create, read, update or delete entities
Team members	Create, read or update entities
Guest	Read

6.3. Comparison with Other Project Management Tools

Microsoft Project has support for complex management tasks such as Gantt charts and resource tracking. Redmine has fewer project management features; with its bug and issue tracking capabilities, the focus is on software projects. In comparison with these two tools, the digital library development management tool has a minimal set of project management capabilities such as tracking user activity and project progress, but its built-in support for describing digital library models makes it well-suited for managing and understanding digital libraries. Table 12 compares these project management tools by feature areas.

Table 12. Microsoft Project vs. Redmine vs. Digital library development management tool

Microsoft Project	**Redmine**	**DL development management tool**
Collaboration features		
• Email Integration • RSS Feed • Team Calendars	• Email Integration • Forums • Issue tracking	• Email Integration (suggested) • Task assignments • Team calendars
Project management features		
• Calendars • Critical Path Method • Events • Gantt charts	• Document management • Gantt charts • Bug tracking	• Events • Timelines • Phases • Tasks • Documents management
Resource Management		
• Costs • Email Addresses • Groups • Import Resources • Materials/Supplies • Resource • Resource Details • Skill Sets • Time Sheets	• Time sheets	• Import resources • User groups • Upload and download resources

7. Conclusions and Future Work

7.1. Conclusions

Motivated by the need for project management solutions, specifically for digital library development, we have developed a tool that helps to manage and sustain developmental plans in a more customized way than the generic project management tools currently on the market. We have shown that this can be done by examining the phases of development a digital library undergoes and identifying which entities require supervision.

We also have shown how users can create and oversee the development of their digital libraries, and can keep track of various project-management tasks, assign roles to project participants, track progress of the project through its phases, manage calendar events and share project resources.

Moreover, we demonstrated how the application of the 5S framework can be used to describe the digital library's streams, structures, spaces, scenarios and societies as well as the project management entities like tasks, documents, phases and events.

We also have proposed some ways of correlating the 5S descriptions of the management system with that of the digital library, for instance, assigning users (described by the 5S project management model as 'societies') to oversee a set of collections (described by 5S digital library management model as 'structures') in the digital library.

We have evaluated the tool with two digital library case studies, CTRnet and national digital library. Both these projects helped identify areas of improvement to make the tool more usable in real-world scenarios and these features.

In summary, this thesis has made practical contributions to the field of digital libraries, which we hope will be a good starting point for further efforts in this direction.

7.2. Future Work

The tool is an ongoing development effort and it currently meets some of the basic needs for digital library project-management. It can be further tailored for digital libraries and many other standard features can be added to it. Some of the future work should include:

- Implementing suggested features in the 5S description of project: This includes ability to link tasks to structures as well as users to tools listed in the spaces section.

- Templates for different types of digital libraries: There are different types of digital libraries catering to different types of content or audience. The tool could be extended to support pre-defined templates that when applied to a digital library project, auto-populate the 5S model of the library or the phases of the project. This would help new users quickly get started with managing their digital library projects.

- Private and public digital libraries: Currently, the tool does not support this distinction, though it is an important feature from a security perspective.

- Import the 5S description from another project: Duplication of effort is reduced by allowing users to populate their 5S description of the digital library by importing from another project in the user's repository.

- Perform user-testing and gather feedback to improve the tool: User-feedback is an important part of development of a tool. Changes can be made to it according to input received from a sample group of users.

- Long-term usage and maintenance: To enable the continued upkeep of the tool, the project sourcecode has been made publicly available at GitHub at https://github.com/nstulasi/DLManagementTool. The hosting and database storage is provided by Heroku for up to 5GB. Depending on the usage and funds available, it can be upgraded to an enterprise cloud platform such as Azure.

- User-guidance: A user manual on the tool has been created and made available along with the source-code on GitHub (see documentation folder) so that first-time users can be guided through the menus and functionalities.

References

[1] V. Bush. As we may think. *Atlantic Monthly*, 176:101–108, July 1945.

[2] J. C.R. Licklider. *Libraries of the Future.* MIT Press, Cambridge, Massachusetts, 1965.

[3] E. Fox. The digital libraries initiative: Update and discussion: Guest editor's introduction to special section. *Bulletin of the American Society of Information Science*, 26(1):7–11, 1999.

[4] D. J.Waters.What Are Digital Libraries? *CLIR Issues*, July/August 1998. http://www.clir.org/pubs/issues/issues04.html [last visited July 4, 2012]. DOI: 10.1300/J105v22n03_09

[5] B. M. Leiner. The Scope of the Digital Library. *DLib Working Group on Digital Library Metrics,* January 1998 http://www.dlib.org/metrics/public/papers/dig-lib-scope.html [last visited October 21, 2013].

[6] Carnegie Mellon University. *Trustworthy Refinement Through Intrusion-Aware Design, Appendix: Glossary*. 2007. http://web.archive.org/web/20070903115947/http:/www.sei.cmu.edu/publications/documents/03.reports/03tr002/03tr002glossary.html [last visited October 21, 2013]

[7] E. A. Fox, M. A. Gonçalves, and R. Shen. Theoretical Foundations for Digital Libraries: The 5S (Societies, Scenarios, Spaces, Structures, Streams). *Synthesis Lectures on Information Concepts, Retrieval, and Services,* page 32, Morgan & Claypool Publishers, San Rafael, CA, July 2012. http://www.morganclaypool.com/doi/abs/10.2200/S00434ED1V01Y201207ICR022 [last visited October 21, 2013] DOI:10.2200/S00434ED1V01Y201207ICR022

[8] Project Management Institute. A Guide to the Project Management Body of Knowledge, page 6, 2004.

[9] M. Collier. Development of a business plan for an international co-operative digital library - The European Library (TEL). *Program: Electronic Library and Information systems.* 38:225-33. 2004. DOI: 10.1108/00330330410566060

[10] W. Janko and M. Hahsler. Collaboration of digital libraries. 2004 http://michael.hahsler.net/stud/done/shustitskiy/diglib_shustitskiy.pdf [last visited October 21, 2013]

[11] M. A. Goncalves. Streams, Structures, Spaces, Scenarios, and Societies (5S): A Formal Digital Library Framework and Its Applications. PhD thesis, Virginia Tech, Blacksburg, VA, 2004. http://scholar.lib.vt.edu/theses/available/etd-12052004-135923/ [last visited

July 4, 2012].

[12] Q. Zhu. 5SGraph: A Modeling Tool for Digital Libraries. Master's Thesis, Dept. of Compute. Sci., Virginia Tech, Blacksburg, 2002. http://scholar.lib.vt.edu/theses/available/etd-11272002-210531/ [last visited July 4, 2012].

[13] K. McNeill-Harman. Foundations of digital libraries: Organizational & management issues. *Proceedings of the American Society for Information Science and Technology.* 39:499-500, November 2002. DOI: 10.1002/meet.1450390182

[14] CTRnet. Crisis, Tragedy, and Recovery Network website. http://www.ctrnet.net [last visited October 21, 2013]

[15] Association of Research Libraries. Definition and Purposes of a Digital Library. 1995. http://yunus.hacettepe.edu.tr/~tonta/courses/fall99/kut655/DL-definition.html [last visited October 21, 2013]

[16] A. Dawson. Building a digital library in 80 days: the Glasgow experience. *Digital Libraries: Policy, Planning and Practice*. http://cdlr.strath.ac.uk/pubs/dawsona/ad200301.htm [last visited October 21, 2013]

[17] N. V. House, M. Butler, and L. Schiff. Needs assessment and evaluation of a digital environmental library: the Berkeley experience. *DL96: First ACM Int Conference on Digital Libraries,* Bethesda, MD, March 20–23, 1996.

[18] M. A. Gonçalves, E. A. Fox, L. T. Watson, and N. A. Kipp. Streams, structures, spaces, scenarios, societies (5S): A formal model for digital libraries. Technical Report TR-03-04, Computer Science, Virginia Tech, Blacksburg, VA, 2003. http://eprints.cs.vt.edu/archive/00000653/ [last visited October 21, 2013]. DOI: 10.1145/984321.984325

[19] T. Mornini. *Here's Why Ruby On Rails Is Hot*. http://articles.businessinsider.com/2011-05-11/tech/30035869_1_ruby-rails-custom-software [last visited October 21, 2013]

[20] Microsoft. Microsoft Project 2012 website. http://www.microsoft.com/project/en-us/project-management.aspx [last visited October 21, 2013]

[21] J. P. Lang. Redmine features 2012. http://www.redmine.org/projects/redmine/wiki/Features [last visited October 21, 2013]

[22] 37signals. Ruby on Rails website. www.rubyonrails.org [last visited October 21, 2013]

[23] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software.* Addison-Wesley Professional. 1994

[24] Indeed.com. Ruby, Rails, Python, PHP, Java Job Trends. 2012. http://www.indeed.com/jobtrends?q=ruby%2C+rails%2C+python%2C+php%2C+java&l=&relative=1 [last visited October 21, 2013]

[25] 37signals. Ruby gems. http://rubygems.org/ [last visited October 21, 2013]

[26] J. French. Formtastic gem. https://github.com/justinfrench/formtastic [last visited October 21, 2013]

[27] A. Patterson, M. Dalessio, C. Nutter, S. Arbeo, P. Mahoney, and Y. Harada. Nokogiri. http://nokogiri.org/ [last visited October 21, 2013]

[28] M. Marohnić. will_paginate gem. https://github.com/mislav/will_paginate/wiki/ [last visited October 21, 2013]

[29] M. Hartl, *Ruby on Rails Tutorial: Learn Rails by Example*. Addison-Wesley. Boston, Massachusetts, USA, 2010.

[30] F. Marinescu. Tim Bray compares intrinsic qualities of Java, Rails, PHP*[]* http://www.infoq.com/news/2006/11/tim-bray-intrinsics [last visited October 21, 2013]

[31] GitHub. GitHub website. https://github.com/ [last visited October 21, 2013]

[32] M. Fowler. *Patterns of Enterprise Application Architecture*. Addison-Wesley. Boston, Massachusetts, USA, 2002.

[33] SQlite. Most Widely Deployed SQL Database. http://www.sqlite.org/mostdeployed.html [last visited October 21, 2013]

[34] PostgreSQL. PostgreSQL website. http://www.postgresql.org [last visited October 21, 2013]

[35] E. Vlist. XML Schema. O'Reilly Media. Sebastopol, CA, 2002.

[36] P. V. Biron and A. Malhotra, XML Schema Part 2: Datatypes. 2004. http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/ [last visited October 21, 2013]

[37] N. Smith. 960 Grid System. http://960.gs/ [last visited October 21, 2013]

[38] Wikipedia. Live Search. http://ajaxpatterns.org/Live_Search [last visited October 21, 2013]

[39] J. Nielsen. Ten Usability Heuristics. 1995. http://www.useit.com/papers/heuristic/heuristic_list.html [last visited October 21, 2013]

[40] 37signals. ActiveRecord::Associations::ClassMethods. http://api.rubyonrails.org/classes/ActiveRecord/Associations/ClassMethods.html [last visited October 21, 2013]

[41] Q. Zhu, M. A. Goncalves, and E. A. Fox. 5SGraph demo: a graphical modeling tool for digital libraries. *Proc. of the 3rd ACM/IEEE-CS Joint Conference on Digital Libraries,* page 385, JCDL'03, Houston, TX, 2003. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.118.7663&rep=rep1&type=pdf [last visited October 21, 2013]

[42] R. Kelapure. Scenario-Based Generation of Digital Library Services. Master's thesis, Virginia Tech Dept. of Computer Science, 2003. http://scholar.lib.vt.edu/theses/available/etd-06182003-055012/ [last visited October 21, 2013].

[43] B. L. Moreira, M. A. Goncalves, A. H. F. Laender, and E. A. Fox. Evaluating Digital Libraries with 5SQual. *Research and Advanced Technology for Digital Libraries,* pages 466-470, Berlin / Heidelberg, UK, 2007. Springer.

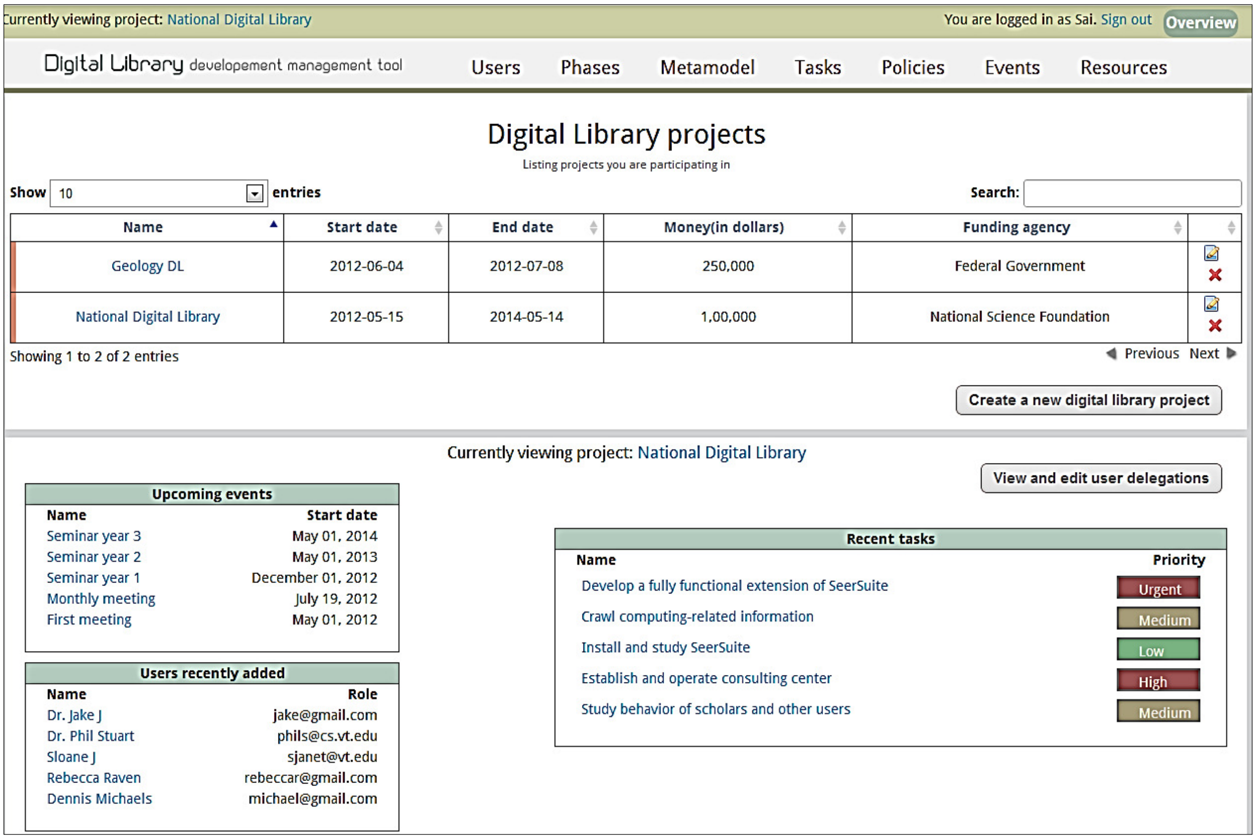
[44] P. T. Baxter. Management Dashboards and Requirements Measurement. Telelogic White Paper, 2006.

[45] M. Bowman and B. Camargo. Digital libraries: the next generation in file system technology. *D-Lib Magazine.* February 1998. http://www.dlib.org/dlib/february98/bowman/02bowman.html [last visited October 21, 2013].

[46] M. A. Gonçalves and E. A. Fox. 5SL: a language for declarative specification and generation of digital libraries. *Proceedings of the 2nd ACM/IEEE-CS Joint Conference on Digital Libraries*, pages 263-272, Portland, Oregon, USA, July 14 - 18, 2002.

[47] Wikipedia. Internet Media Type. http://en.wikipedia.org/wiki/Internet_media_type [last visited October 21, 2013]

Appendix A: Screenshots of the Tool- A Case Study of the National Digital Library.


Currently viewing project: National Digital Library
You are logged in as Sai. Sign out
Overview
Digital Library developement management tool
Users
Phases
Metamodel
Tasks
Policies
Events
Resources

Digital Library projects

Listing projects you are participating in

Show 10 entries

Search:

Name	Start date	End date	Money(in dollars)	Funding agency	
Geology DL	2012-06-04	2012-07-08	250,000	Federal Government	✎ ✖
National Digital Library	2012-05-15	2014-05-14	1,00,000	National Science Foundation	✎ ✖

Showing 1 to 2 of 2 entries

◀ Previous Next ▶
Create a new digital library project

Currently viewing project: National Digital Library

Upcoming events

Name	Start date
Seminar year 3	May 01, 2014
Seminar year 2	May 01, 2013
Seminar year 1	December 01, 2012
Monthly meeting	July 19, 2012
First meeting	May 01, 2012

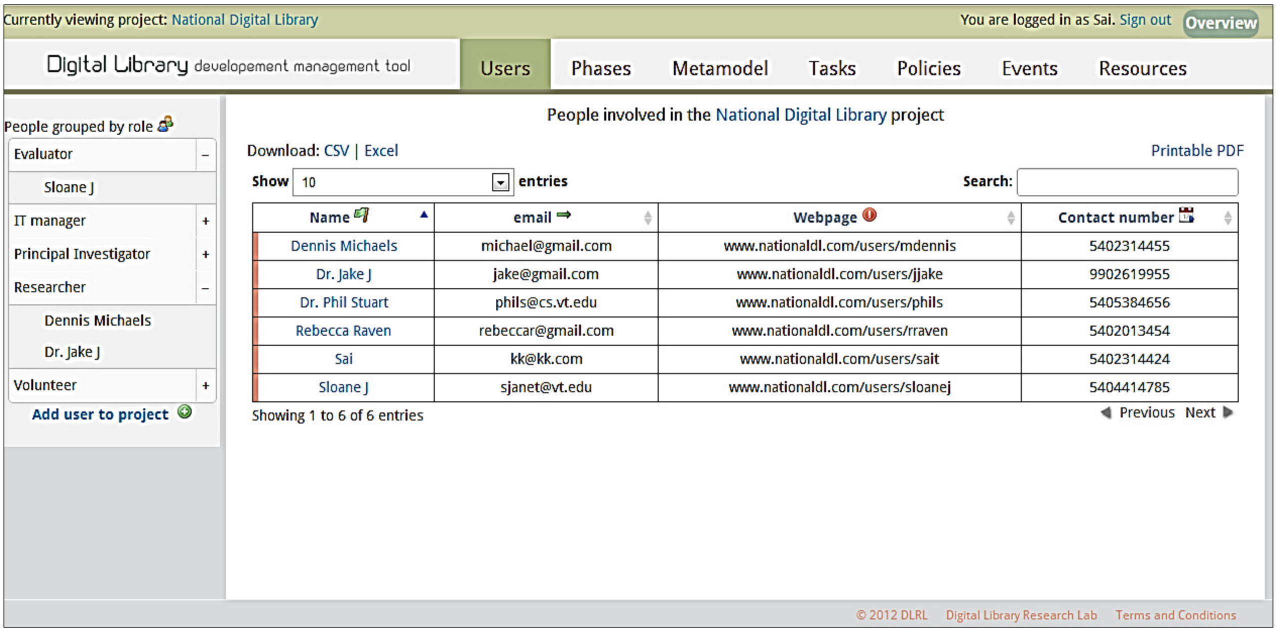
Users recently added

Name	Role
Dr. Jake J	jake@gmail.com
Dr. Phil Stuart	phils@cs.vt.edu
Sloane J	sjanet@vt.edu
Rebecca Raven	rebeccar@gmail.com
Dennis Michaels	michael@gmail.com

Recent tasks

Name	Priority
Develop a fully functional extension of SeerSuite	Urgent
Crawl computing-related information	Medium
Install and study SeerSuite	Low
Establish and operate consulting center	High
Study behavior of scholars and other users	Medium

Figure 46. Projects overview page showing a listing of the users' projects and the dashboard below with latest information about the project

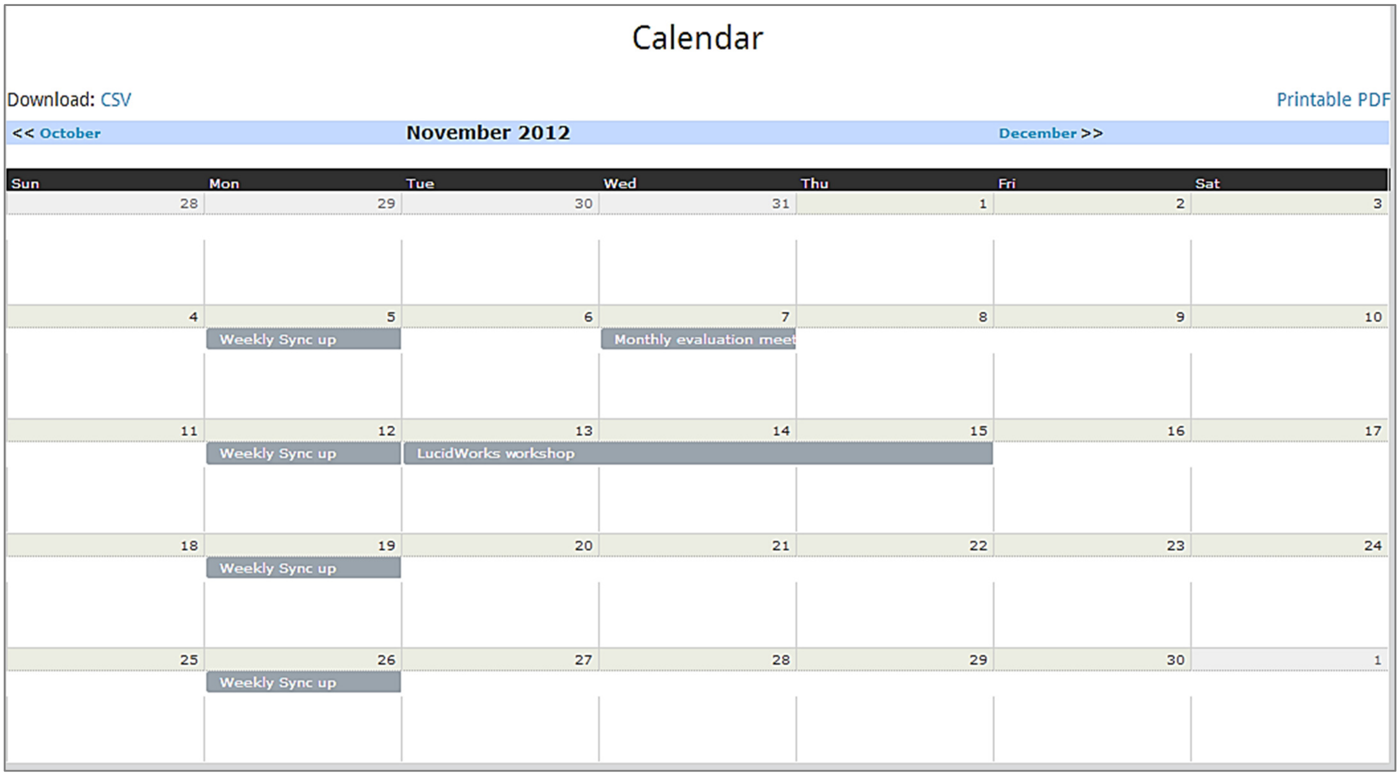


People involved in the National Digital Library project

Name	email	Webpage	Contact number
Dennis Michaels	michael@gmail.com	www.nationaldl.com/users/mdennis	5402314455
Dr. Jake J	jake@gmail.com	www.nationaldl.com/users/jjake	9902619955
Dr. Phil Stuart	phils@cs.vt.edu	www.nationaldl.com/users/phils	5405384656
Rebecca Raven	rebeccar@gmail.com	www.nationaldl.com/users/rraven	5402013454
Sai	kk@kk.com	www.nationaldl.com/users/sait	5402314424
Sloane J	sjanet@vt.edu	www.nationaldl.com/users/sloanej	5404414785

Showing 1 to 6 of 6 entries

Figure 47. The user index page showing a listing of all the users in the project



Calendar

Download: CSV

Printable PDF

<< October November 2012 December >>

Sun	Mon	Tue	Wed	Thu	Fri	Sat
28	29	30	31	1	2	3
4	5 Weekly Sync up	6	7 Monthly evaluation meet	8	9	10
11	12 Weekly Sync up	13 LucidWorks workshop	14	15	16	17
18	19 Weekly Sync up	20	21	22	23	24
25	26 Weekly Sync up	27	28	29	30	1

Figure 48. Calendar view of the project's events

Appendix B: XML Schema for Tasks, Users and Phases

XML Schema for Tasks

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schemaxmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
<xs:element name="root">
<xs:complexType>
<xs:sequence>
<xs:element maxOccurs="unbounded" ref="task"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="task">
<xs:complexType>
<xs:sequence>
<xs:element ref="name"/>
<xs:element ref="status"/>
<xs:element ref="priority"/>
<xs:element ref="start_date"/>
<xs:element ref="end_date"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="name">
<xs:simpleType>
<xs:restriction base="xs:string">
<xs:pattern value="([a-zA-z0-9 ])*"/>
<xs:minLength value="3"/>
<xs:whiteSpace value="collapse"/>
</xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element name="status">
<xs:simpleType>
<xs:restriction base="xs:string">
<xs:enumeration value="Not Started"/>
```

```
<xs:enumeration value="In Progress"/>
<xs:enumeration value="Nearing Completion"/>
<xs:enumeration value="Completed"/>
<xs:enumeration value="Undefined"/>
</xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element name="priority">
<xs:simpleType>
<xs:restriction base="xs:string">
<xs:enumeration value="Lowest"/>
<xs:enumeration value="Low"/>
<xs:enumeration value="Medium"/>
<xs:enumeration value="High"/>
<xs:enumeration value="Urgent"/>
<xs:enumeration value="Undefined"/>
</xs:restriction>
</xs:simpleType></xs:element>
<xs:element name="start_date" type="xs:date"/>
<xs:element name="end_date" type="xs:date"/>
</xs:schema>
```

XML Schema for Users

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schemaxmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
<xs:element name="root">
<xs:complexType>
<xs:sequence>
<xs:element maxOccurs="unbounded" ref="user"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="user">
<xs:complexType>
<xs:sequence>
<xs:element ref="name"/>
```

```
<xs:element ref="email"/>
<xs:element ref="webpage"/>
<xs:element ref="number"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="name">
<xs:simpleType>
<xs:restriction base="xs:string">
<xs:pattern value="([a-zA-z. ])*"/>
<xs:minLength value="3"/>
<xs:whiteSpace value="collapse"/>
</xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element name="email">
<xs:simpleType>
<xs:restriction base="xs:string">
<xs:pattern value="[^@]+@[^\.]+\..+"/>
</xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element name="webpage">
<xs:simpleType>
<xs:restriction base="xs:anyURI">
<xs:minLength value="3"/>
<xs:whiteSpace value="collapse"/>
</xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element name="number">
<xs:simpleType>
<xs:restriction base="xs:string">
<xs:pattern value="\+\d{2}\s\d*\s*\d*\s*\d*"/>
<xs:minLength value="3"/>
<xs:whiteSpace value="collapse"/>
</xs:restriction>
</xs:simpleType>
```

```
</xs:element>
</xs:schema>
```

XML Schema for Phases

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schemaxmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
<xs:element name="root">
<xs:complexType>
<xs:sequence>
<xs:element maxOccurs="unbounded" ref="phase"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="phase">
<xs:complexType>
<xs:sequence>
<xs:element ref="name"/>
<xs:element ref="content"/>
<xs:element ref="start_date"/>
<xs:element ref="end_date"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="name">
<xs:simpleType>
<xs:restriction base="xs:string">
<xs:pattern value="([a-zA-z. ])*"/>
<xs:minLength value="3"/>
<xs:whiteSpace value="collapse"/>
</xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element name="content" type="xs:string"/>
<xs:element name="start_date" type="xs:date"/>
<xs:element name="end_date" type="xs:date"/>
</xs:schema>
```