A LOOSELY COUPLED ARCHITECTURE FOR DIGITAL LIBRARIES

The Phronesis Case

Juan C Lavariega, Andan Salinas, David Garza, Lorena Gomez, Martha Sordia
Centro de Investigacion ene Informatica, ITESM-Campus Monterrey, Mexico

Keywords: Digital libraries, Software architecture, Information systems

Abstract: Phronesis is a software tool for the creation and administration of digital libraries that can be geographically distributed and that are accessible over the WWW. Phronesis was developed by using open source software with the intention to make the project accessible to other developers, who can improve its functionality. However, one of the major drawbacks in Phronesis was its data centric architecture and the highly coupled subsystems which made hard to maintain or to add new functionality. This paper addresses the problems with the old data centric Phronesis architecture. Throughout the paper we discuss the functionality provided by the subsystems, and present a loosely coupled architecture for digital libraries. The approach presented here follows the style of services oriented architectures (SOA). The SOA for Phronesis is a framework that provides services for the submission, indexing and compression of documents. Phronesis SOA is organized into layers of functionality that favour maintenance, reuse, and testing of the entire project. Also the SOA increases Phronesis performance and availability..

1 INTRODUCTION

A Digital Library (DL) (Fox,2002) is an organized collection of documents stored in digital format (i.e., text, image, video or any combination of these formats). Digital libraries must also face challenges for integration of diverse information about the documents (i.e. metadata), and interoperation with other digital libraries. A digital library must provide services for *submission, indexing, classification, storage, searching, retrieval,* and *administration* of the digital documents within its collection or collections.

Phronesis project (Garza, 2003) started with the goal of contributing to the research and development of digital libraries technology. The main result of the Phronesis project is a freely-available single-system software tool that can be used for the creation of distributed digital libraries on the Internet. The Phronesis system supports storage and retrieval of any digital documents (video, software, text, etc), search and retrieval for text-based documents, support for indexing , searching and retrieving documents written in English or Spanish, use and administration of the library via WWW over the Internet, interoperability with digital libraries that use the Z39.50 protocol and stemmer functionality.

Phronesis follows a client-server model with centralized data, in which all of the digital library functionality is located in one server. The document repository or digital collection resides in one site that is accessed remotely by clients through a web interface. The interface allows users to perform different tasks such as: creation of digital documents; classification and indexing of the documents within the same collection; search and retrieval based on keywords or metadata, administration; and access control. This architecture is in essence monolithic and all the components are highly coupled with each other. Adding new functionality to Phronesis became a major issue to consider when new requirements arise.

For this reason, a new Phronesis' architecture with services as components organized into several abstraction layers was developed. This idea comes from two architectural styles: Service Oriented Architecture (SOA) and Layered Architecture. These architectural styles provide the capability to move tasks to specialized software servers and the scalability that allows the incremental and structured growth on the functionality that a digital library requires.

C Lavariega J., Salinas A., Garza D., Gomez L. and Sordia M. (2005).
A LOOSELY COUPLED ARCHITECTURE FOR DIGITAL LIBRARIES - The Phronesis Case.
In *Proceedings of the Seventh International Conference on Enterprise Information Systems*, pages 495-498
DOI: 10.5220/0002533204950498

2 PHRONESIS DATA CENTRIC ARCHITECTURE

Phronesis follows the client server architecture, with the server as the main component composed of several internal subcomponents for search, retrieval, indexing and managing of documents in the digital library. Clients are World Wide Web interfaces using HTTP, HTML and PERL Common Gateway Interfaces (CGI's) technology.

Phronesis clients are users accessing the system for performing actions such document search and retrieval as well as document submission. Three types of users are presented in Phronesis:

Collection Contributors. They are users that have the proper permissions to submit documents to the collection.

Administrators. They are users that maintain a Phronesis server.

Patrons. They are users who access the server to search and retrieve full documents.

The server, key component of the system, performs the following tasks: administration, access control, physical storage of documents, indexing, local and distributed search and retrieval. Functionality for document storage and retrieval is based on MG (Witten 1999), a powerful research tool for the compression, indexing and retrieval of textual documents. We have extended the MG system in order to provide all the desired functionality in Phronesis.

Phronesis supports search and retrieval of English and Spanish documents. The user interface is also available in both languages. The server implements five different types of full-text document and/or metadata search. The search query can include diacritic characters common in the Spanish language. When searching words that contain diacritic characters, the server tolerates simple mistakes common in Spanish language, such as the omission of an accent. For example, a search for the word computing in Spanish will be performed using the keyword "computación". If the keyword "computacion" (accent omitted) is used, the query will find the same documents as the previous one. A stemmer algorithm to support Spanish language is also available as part of the system. A single Phronesis system can interact not only with other Phronesis systems but also provides interoperability support for Z39.50 based libraries and Open Archives Initiative based libraries.

The current components of the data-centric architecture of a Phronesis server are a set of subsystems working together to satisfy services of a digital library. Since all of these services are tightly-coupled, making improvements is a time consuming process that requires a good amount of knowledge about the internals of the system.

As a result, the Phronesis system has become a complex, monolithic piece of software, hard to maintain, with no flexibility to easily evolve. A clear sign of the inherent problems with the current Phronesis' architecture is the time that it takes to run a test case compared with the time it took to run the same test case in previous versions. Also, it takes longer to add new functionality. The differences in time are because Phronesis' components are highly coupled and with a poor cohesion. This complexity has an impact in the quality assurance process since fixing a bug in one component may introduce problems in another component. Therefore, a redesign of the architecture was highly needed in order to support future improvements of the system.

2.1 Analysis of the Data Centric Architecture

Phronesis' subsystems are not implemented as independent components with well-defined boundaries. Phronesis is a highly integrated set of programs and tools that interact by means of the shared data in the central repository.

Based on their functionality (Garza, 2003) each program or tool can be further classified as follows:

Document Searching Subsystem. This subsystem allows the searching of documents in different and distributed Phronesis' repositories.

Document Browsing Subsystem. This subsystem includes all the programs for visualizing the documents stored in the repository. Visualization is based on predefined categories such as document title, year of publication, or author's name.

Document and Library Builder Subsystem. This subsystem groups the programs for document processing and index creation that is required during document searching.

Document Storage Subsystem. This subsystem includes the programs and file structures required to assign unique identifiers to documents. It also includes the programs to save and to retrieve documents, and to do the preprocessing needed prior to the indexing of documents.

Management Subsystem. This subsystem includes programs for the digital library configuration.

Z39.50 Interoperability Subsystem. This subsystem includes all the programs that allow Phronesis to perform queries to/from libraries that support Z39.50 protocol

According to (Bass,1998), the following criteria allow to determine the strengths and weaknesses of different architectural styles in software.

Maintainability. This parameter indicates the degree of complexity to make changes in the software as a result of new user's requirements.

Reusability. This criterion indicates if the software product is designed to reuse components from other products, or if its components can be reused in other products.

Ease of Testing. This parameter indicates the degree of difficulty to test a software component. It also determines the complexity to detect defects in the software.

Performance. This parameter is related to the system's response time, or the number of events processed during a time period. Performance also is related to the communication costs (in time) between different modules of the system.

Availability. This parameter measures the time in which the system is operational.

With these criteria, we can evaluate not only the software behaviour and functionality (*Performance* and *Availability*), but also characteristics about the software lifecycle (*Maintainability*, *Reusability* and *Ease of Testing*). All of these parameters were used to evaluate the current data centric architecture and the new service oriented architecture of Phronesis

2.2 Opportunity and Impact

Following is a summary of the opportunity areas and the impact in the original data centric architecture for Phronesis.

Subsystem: Searching

Area of Opportunity: There is not a clear separation between searching requests and the results interface

Impact *Maintenance* is affected because code from interface and code from searching process are mixed which produces program that are difficult to modify. New errors can be introduced with each change. Also without a clear interface it is difficult to *interoperate* with other DLs

Subsystem: Browsing

Area of Opportunity: Each time the browser is invoked all of the metadata components are accessed, which makes the process more time consuming.

Impact *Performance* application deteriorates because waiting time for construction in the browsing menu increases as the collection becomes bigger. **Maintenance** is difficult in the interfaces that present results.

Subsystem: Storage

Area of Opportunity: Since there are several subsystems that perform operations over the same data, subsystems are dependent, which makes more complex any change on an individual subsystem

Impact *Maintenance* and *testing* are affected due to strongly coupled data between the programs. Reuse is also affected because it is not possible for a storage subsystem to store documents from multiple collections.

Subsystem: Management

Area of Opportunity: Administration programs are completely duplicated for different language versions (Spanish and English)

Impact *Reuse* and *Maintenance* are affected because administrative processes have to be duplicated for each language version of the user interface.

Subsystem: Interoperability

Area of Opportunity: There is duplicity in the metadata and indexes, which implies a big effort to keep consistency for that information in different versions.

Impact *Maintenance* is affected because it is required to assign resources for servers that require communication.

Subsystem: Builder

Area of Opportunity: The process that extracts the text from documents can become a bottleneck in the system.

Impact *Availability* is affected. As the number of documents increases, the index reconstruction process takes longer limiting the availability of the system because during index construction time, searches cannot be performed in the library.

3 PHRONESIS LOOSELY COUPLED ARCHITECTURE

A Service Oriented Architecture is an architectural style for software products whose objective is to have a loosely-coupled interaction among components in the system. These components are in essence a collection of services communicating with each other. Communication between components may be a simple data interchange, or one or more services coordinating one activity. Each service is a well-defined function, self-contained, and independent of the state of another service. After analysing the functions each subsystem provides, six basic services were identified:.

Searching and Indexing Service. This component maintains communication with most of the other services and provides basic services needed for location of documents within a collection. This service also builds documents' indexes and processes searches requested by users that invoked parallel searching or interoperability services.

Storage Service. This service provides functionality for storing and retrieving any

document in the collection. This service is usually accessed by the searching service in order to index new documents; and by the user interface, so users can get any document.

Parallel Searching Service. This service replicates a searching request in several repositories at once and generates a global result list.

Configuration Service. This service manages information about the services and resources that form the digital library.

Interoperability Service. This service provides a wrapper for accessing Phronesis' services by using Z39.50 protocol. It is possible to create communication services with other digital libraries.

User Service. This service manages and maintains user information such as access control data and preferences. It also authenticates users accessing Phronesis.

The new Phronesis Loosely Coupled Architecture, is based on a layered architectural style in which each layer groups a set of services that together provide the functionality of the digital library.

Presentation Logic Layer. This layer provides the user interface. Programs at this level work with the outputs resulting from programs at the business layer and transform such results into a format useful for users. This layer includes menu lists, forms for capturing data, and programs for transforming XML into HTML documents.

Business Logic Layer. This layer includes the programs implementing business rules. At this level, programs coordinate the flow of events that occur during the operation of the digital library.

Service Access Layer. This layer is an application program interface that allows access to service function in the digital library. This layer hides the existence of different specialized services such as searching, users' administration and configuration.

Service Layer. This layer consists of the services for data management in the digital library. Each service performs specific and limited functionality. These services are used as building blocks by upper layers.

Proxy Layer. This layer is used for interaction with other applications. The proxy layer works as an external application accessing the services of the Business logic layer. The proxy layer allows sharing the digital library functionality by using other communication protocols.

4 CONCLUSIONS

We have presented two different architectures of Phronesis, an open, self-contained software product that provides digital library services.

The first architecture, called a data centric architecture, provides the functionality required. However, the integration of new user requirements is a process that becomes more complex as the system grows. Because of the complexity of the software, the performance also is affected. To avoid all these problems, a new architecture based on services was developed. The new architecture contains independent, self-contained modules that performed a well-defined function. As a result, the software produced should have properties such as maintainability, reusability, testability and availability

REFERENCES

Bass, L., et al, 1998. *Software Architectures in Practice,* . Addison Wesley..

Fox, E.A,2002, Overview of Digital Libraries, *Proceedings of the second ACM/IEEE-CS Joint Conference on Digital Libraries*, ACM press.

Garza-Salazar, D et al 2003 “Information Retrieval and Administration of Distributed Documents in Internet,” in *Knowledge-Based Information Retrieval and Filtering from the Web* Edited by W. Abramowicz, Kluwer Academic Publishers

Witten, I.H.; et al 1999 *Managing Gigabytes Compressing and Indexing Documents and Images, 2nd. Edition,* Morgan Kauffman Publishers,