The Architecture of the Web Service for Automated Storage of Scientific Papers

Miloš Radmanović

Abstract -- **The objective of this paper is to present a solution for the architecture of the Web service for automated storage of scientific papers, applied at the Computational Intelligence and Information Technologies Archive website. This Web service is based on Active Server Pages (ASP) technology and three-tier client-server architecture. The architecture of this Web service is modeled with different types of system model. Different types of system model are based on different approaches to abstraction. This paper includes tree-tier client-server overview, system structuring and object-oriented system design using Unified Modeling Language (UML) extension for Web applications. This architecture made this Web service applicable in the modern Web trends.**

Keywords — **Web service, Web application, three-tier client-server, architecture, system structuring, ASP, UML**

I. INTRODUCTION

Web applications are becoming more and more popular. This is in part due to the rapid deployment of the tools and technologies for developing them. But mostly because system designers are recognizing the situations where Web applications have very significant advantages over traditional applications [5].

To date the focus of web application development has been the tools. Little attention has been paid to the development process. Current development environments make it so easy to produce simple Web applications that they have the unfortunate side effect of encouraging us to develop and evolve applications in the absence of serious analysis and design. Any system with non-trivial complexity needs to be designed and modeled. A secondary goal of this paper is to stress the need for proper modeling. Unfortunately modeling web applications is not obvious [5].

In this paper I report 4 month development of Web architectures and Web application for automated storage of the scientific papers as a part of the Computational Intelligence and Information Technologies Archive website placed on Faculty of Electronic Engineering Niš. Most of my experience with Web applications comes from my use of Microsoft's Active Server Page technology. I have made a very strong attempt to make the discussions in this paper as generic as possible.

M. Radmanović, is with the Faculty of Electronic Engineering, University of Niš, Beogradska 14, 18000 Niš, Yugoslavia (e-mail: milos@elfak.ni.ac.yu).

The organization of Web service for automated storage of scientific paper gives the possibility for: data entry and search simplicity, scientific papers classification, multi relations of the members, accounts, authors, scientific papers, sections and Workshops, automatic generation of news and events and automatic storage of PDF (Portable Document Format) documents (scientific papers).

This paper begin with brief explanation of three-tier client-server architecture and high level view of the Web service for automated storage of scientific paper. In Section 3, I explain system structuring and present structural model of the architecture for Web service for automated storage of scientific papers. In Section 4, I discuss about UML extension for Web applications and I model some elements of Web service using UML extension for Web applications. Conclusions are given in Section 5.

II. THREE-TIER CLIENT-SERVER ARCHITECTURE OF THE SYSTEM

The term client/server was first used in the 1980s in reference to personal computers (PCs) on a network. The actual client/server model started gaining acceptance in the 1980s. The client/server software architecture is a versatile, message-based and modular infrastructure that is intended to improve usability, flexibility, interoperability and scalability, as compared to centralized, mainframe, time sharing computing. A client is defined as a requester of services and a server is defined as the provider of services [1].

The three-tier software architecture, also known as three layer architectures, emerged in the 1990s to overcome a limitation of the two tier architecture. The third tier (middle tier server) is between the user interface (client) and the data management where the business logic and rules are executed and can accommodate hundreds of users (as compared to only 100 users with the two-tier architecture) by providing functions such as queuing, application execution, and database staging.

A three tier distributed client/server architecture, as shown in Figure 1, includes a user system interface top tier where user services (such as session, text input, dialog and display management) reside.

The third tier provides database management functionality and is dedicated to data and file services that can be optimized without using any proprietary database management system

languages. The data management component ensures that the data is consistent throughout the distributed environment through the use of features such as data locking, consistency, and replication.

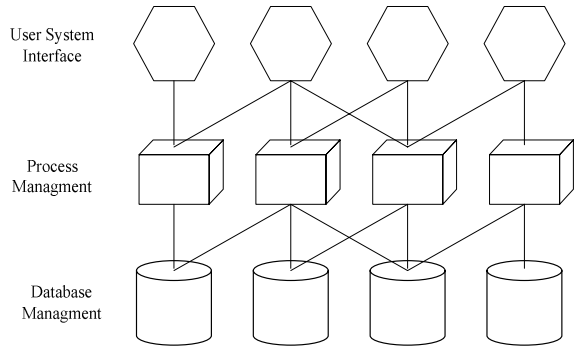

Figure 1: Three tier distributed client/server architecture depiction

It should be noted that connectivity between tiers can be dynamically changed depending upon the user's request for data and services. The middle tier provides process management services (such as process development, process enactment, process monitoring, and process resource) that are shared by multiple applications. The middle tier server (also referred to as the application server) improves performance, flexibility, maintainability, reusability, and scalability by centralizing process logic. Centralized process logic makes administration and change management easier by localizing system functionality so that changes must only be written once and placed on the middle tier server to be available throughout the systems.

Sometimes, the middle tier is divided in two or more unit with different functions, in these cases the architecture is often referred as multi layer. This is the case, for example, of some Internet applications. These applications typically have light clients written in HTML and application servers written in C++ or Java, the gap between these two layers is too big to link them together. Instead, there is an intermediate layer (web server) implemented in a scripting language. This layer receives requests from the Internet clients and generates html using the services provided by the business layer. This additional layer provides further isolation between the application layout and the application logic [2].

This section provides a high level view of the Web service for automated storage of scientific paper and the system configuration [3]. The system uses three-tier client-server architecture.

The system will consist of a client component and two server components as illustrated in Figure 2.

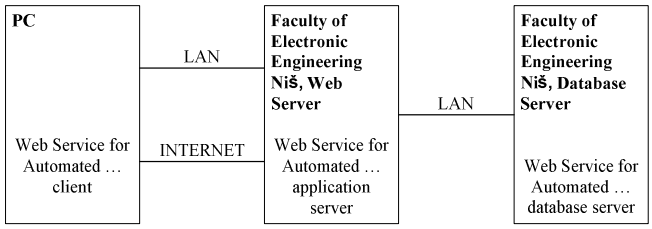

Figure 2: The Web Service for Automated Storage of Scientific Paper Overview

The Web application (first server component) resides on the Faculty of Electronic Engineering Niš Web Server. The Web application component must interface with the database system (second server component) on the Faculty of Electronic Engineering Niš Database Server. This interface is supported by an existing Open SQL Interface. The client component resides on a personal computer. Once the client component (Web browser) is installed on the PC, the user may access the Web Service for Automated Storage of Scientific Paper from the PC through the faculty LAN or Internet. A valid user name and password must be entered in order for full access to be granted.

III. SYSTEM STRUCTURING

As a part of the system requirements and design activity, the system has to be modeled as a set of components and relationship between these components. These are normally illustrated graphically in a system architecture model that gives the reader an overview of the system organization.

The first phase of the architectural design activity is usually concerned with decomposing a system into a set of interacting sub-systems. At its most abstract level, an architectural design may be depicted as a block diagram where each box in the diagram represents a sub-system. Boxes within boxes indicate that the sub-system has itself been decomposed to sub-systems. Arrows mean that data and\or control is passed from sub-system to sub-system in the direction of the arrows. An architectural block diagram presents an overview of the system structure. It is generally understandable to the various engineers who may be involved in the system development process [4].

Figure 3 is a structural model of the architecture for Web service for automated storage of scientific papers.

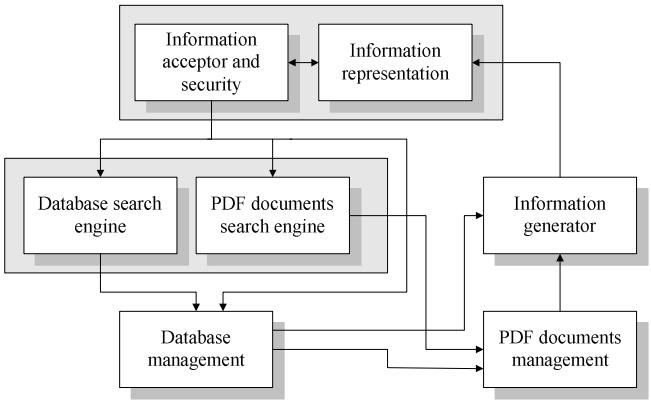

Figure 3: Block diagram of the Web service for automated storage of scientific paper

This system can search, manage, update and present scientific paper's archive. It uses information acceptor and security subsystem to enable user's data entry, authorized access and to identify the type of information. Depending of the type of information, system uses Database search engine subsystem, PDF documents search engine subsystem or Database management subsystem. Database search engine transforms information for SQL queries. PDF documents search engine include PDF Indexing Service. Database management and PDF documents management return sets of data from witch information generator subsystem prepare information for displaying. Information representation subsystem generate User interface.

IV. Modeling Web service (system) with UML

UML is the standard modeling language for modeling software intensive systems. Using UML for the analysis and design models, however, raises some questions when modeling pages and hyperlinks. UML, out of the box, is not a perfect fit.

The creators of UML recognized that standard UML is not a perfect fit for all types of applications, and so have defined a formal way to extend it to meet the needs of these special situations. Web applications represent one of these situations, and this part of section is devoted to explaining the UML Extension for Web Applications.

The UML Extension for Web Applications is a document that defines how web applications can be modeled using UML. The document defines a set of stereotypes, tagged values, and constraints. Stereotypes are new modeling elements. They in sense increase the vocabulary of the UML language. Stereotypes can specify special icons to use in diagrams, to help identify them as "special" elements in model diagrams. Tagged values are simply extra information that can be tacked onto a model element. A collection of these definitions, packaged together is an extension. The UML extension for web applications is designed to define a mechanism by which elements of web applications, most notably; web pages, client scripts, applets, and ActiveX controls can be incorporated into the UML models of the rest of the system [6], [7].

This part of section is devoted to model some elements of Web service for automated storage of scientific paper using UML extension for web application.

Figure 4 is showing associations that indicate that one client page is requesting a link to other ASP pages. A «targeted link» stereotype is applied to associations between client pages and targets that they interact with.

Figure 5 is showing the structure of Search page. Search page can work into three different search modes: basic search, search by author and advanced search. Each mode includes proper form. A form identifies a specific web page (almost always one with a server page stereotype) to accept and process data submitted with the form. A «submits» association stereotype represents the relationship between a form and the web page that processes it.

Figure 6 is showing the structure of Login page. Login page include form for user name and password.

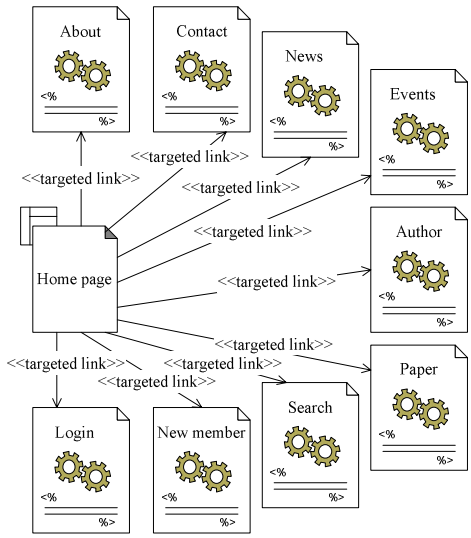

Figure 4: Model of Home page

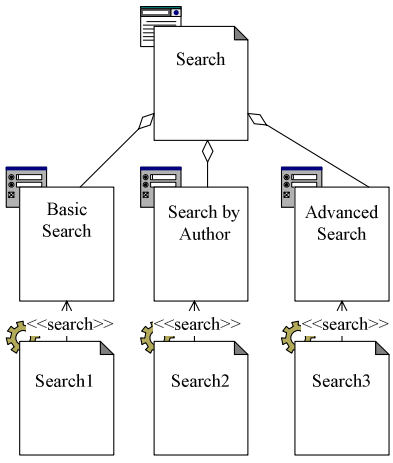

Figure 5: Model of Search page

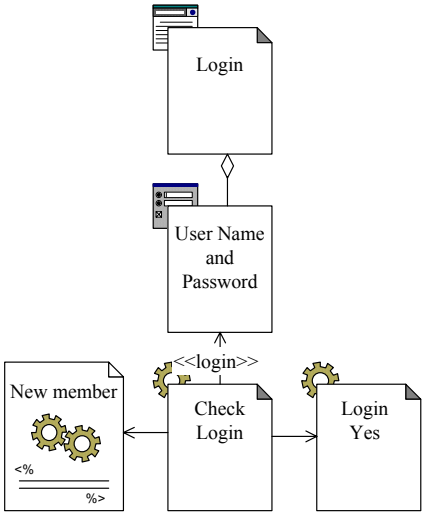

Figure 6: Model of Login page

V. CONCLUSION

In this paper it is presented a solution for the architecture of the Web service for automated storage of scientific papers. It is shown three types of system model: tree-tier client-server overview, system structuring model and object-oriented model using UML extension for Web applications.

Using traditional approach at the developing Web applications, often missing some features and the quality is not as customers expecting for. With the assumptions that modeling is important, we should model the artifacts of a system.

REFERENCES

[1] "Client-Server Software Architectures--An Overview", http://www.sei.cmu.edu/str/descriptions/clientserver_body.html
[2] "Three Tier Software Architectures", http://www.sei.cmu.edu/str/descriptions/threetier_body.html
[3] Sue Gamble, "C-Registration System Vision Document", Wylie College Information Systems, 1998.
[4] Ian Sommervile, "Software Engineering", Pearson Education, 2001 (6th ed.).
[5] Jim Conallen , "Modeling Web Applications with UML", Conallen, Inc, 1998.
[6] Jim Conallen , "The UML Extension for Web Applications", Conallen, Inc, 1998.
[7] Jim Conallen , "UML Extension for Web Applications 0.91", Conallen, Inc, 1998.