

WEB APPLICATIONS ARCHITECTURE

Nicoleta DAVID*, Claudia CÂRSTEA*

*"George Baritiu" University of Brasov

Abstract: Web Applications are software which can be accessed via the Internet or an Intranet using a web browser. There are five architectural views that are fully described in this paper. The approach of Web Engineering should have an associated architecture, which takes into account the special features and needs for the development of Web Applications. A software architecture determines how system components are identified and the interaction between them and the interface protocols for communication. Software is structured as layers in a single computer or as services offered and requested between processes located in the same or different computers. Regarding the systems architectures, the main types of architectural models are the Client-server model and Services provided by multiple servers. The Web Application can be securized by using authentication, by restricting access to certain resources in the Web Application, or by using security calls in the servlet code.

Keywords: Web Application, software architecture.

1. INTRODUCTION

Web Applications are software which can be accessed via the Internet or an Intranet using a web browser. Websites such as *Amazon.com* and *eBay*, and software such as *Blogs*, *Content Management Systems*, *Online Community Forums*, and even your *Online Banking System* are all good examples of services streamed through Web Applications.

A Web Application can also be used as an alternative to installing individually distributed software on desktop computers (i.e. the Desktop Mail or Office Suite applications). Many people now use web-based email such as *Gmail*, *Hotmail*, etc., instead of desktop email applications, such as *Outlook Express* or *Thunderbird*. Services such as *Google Maps*, *Google Calendar*, or *Google Spreadsheets* also belong to this group.

Web Application Architecture is in fact a **specialized form of Software Engineering**. It is the science and craft of designing and implementing computer software that runs on web servers, instead of running directly on desktop computers or laptops.

It also involves significant amount of

Business Process Study and Visual Design for designing Graphical User Interfaces, therefore a **Web Application Architect** is often someone experienced in multiple fields such as Computer Science, Marketing, and Visual Design.

2. ARCHITECTURAL VIEWS

Complex systems cannot be understood from just a single viewpoint. The next diagram illustrates an architectural view of web applications composed by five sections:

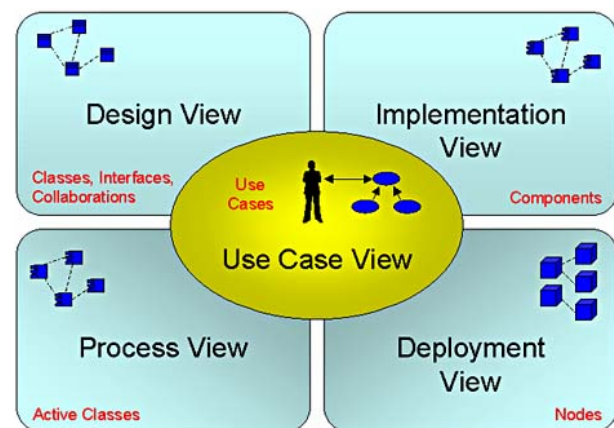


Fig. 1 Architectural views

A system's *design view* encompasses the classes, interfaces, and collaborations that form the vocabulary of the problem and its solution. This view primarily supports the functional requirements of the system, meaning the services that the system should provide to its end-users.

The *process view* of a system encompasses the threads and processes that form the system's concurrency and synchronization mechanisms. This view primarily addresses the performance, scalability, and throughput of the system.

The *implementation view* of a system encompasses the components and files that are used to assemble and release the physical system. This view primarily addresses the configuration management of the system's releases. The releases are comprised of somewhat independent components and files that can be assembled in various ways to produce a running system.

The *deployment view* of a system encompasses the nodes that form the system's hardware topology on which the system executes. This view primarily addresses the distribution, delivery, and installation of the parts that make up the physical system.

The development view for Web Applications must highlight additional details such as:

- The link structure of the application pages;
- User's session management techniques;
- Application page generation technology.

Web Applications use the HTTP protocol as their communication medium. Since the HTTP protocol is a stateless protocol, the Web Application has to use techniques such as Cookies or hidden fields in a web page to store the state for a particular session.

The *use case view* of a system encompasses the use cases that describe the behavior of the system as seen by its end-users, analysts, and testers. This view exists to specify the forces that shape the system's architecture.

The *goal of the WWW project* is to establish a shared information space through which people and machines could communicate. People intending to use this system were

located around the world. The information to be shared was ranging over many different types, such as text documents, pictures, etc. The challenge was to build a system that would provide a universally consistent interface to this structured information.

The Web's information sources would be distributed across the global internet. Therefore the architecture needed to minimize the interactions with the network.

The architecture had some main concerns:

- How to deal with the latency that comes from the network interactions
- Scalability, since some resources would be particularly newsworthy and lead to sudden spikes in access attempts
- How to introduce a new set of functionality to an architecture that is already widely deployed

This motivates the development of the Representative State Transfer (REST) architecture.

This architecture is an abstraction of the architectural elements within a distributed hyper media system.

It has three classes of architectural elements:

1. Data elements. In REST all data is moved to the location where it will be used.

The components communicate by transferring the a representation of the data in a format matching one of an evolving set of standard data types, selected dynamically based on the capabilities or desires of the recipient and the nature of the data.

2. Processing elements. The REST components are typed by their roles in an overall application.

User agents use a client to initiate a request and become the recipient of the response. An *origin server* is the source for representations of its resources. Other types of components are gateways and proxies.

3. Connecting elements. REST uses various connector types to encapsulate the activities of resources and transferring resource representations. All interactions are stateless. Primary connector types are client and server. Other types of connectors are cache, resolver and tunnel.

3. CATEGORIES OF WEB APPLICATIONS

Web applications are not limited to one type of application. They can range from simple static web pages to sophisticated applications.

Different categories of web applications are grouped together according to their data and control complexity:

1. *Brochure Web Applications*: this is the first generation of Web Applications. They are composed of static web pages and tend not to have much programming logic in them. When developing them the focus is on content development and the layout of graphics and text.

Examples are personal web pages which contains their resume and personal information or web pages about a company's product.

2. *Service oriented applications*: These sites are offering a service to web users. Service oriented applications contain the programming logic needed to implement the service. The layout of the data is often a secondary concern. During maintenance the developers need a good understanding of the control logic.

Examples are web mail services or online word-processing systems.

3. *Data intensive applications*: These are sites that provide an interface to browse and query large amount of data. The main emphasis in these applications is on the data, with minimal amount of programming logic involved. During maintenance the developers need a good understanding of the data flow.

An example of this application type is online library catalogues.

4. *Information system applications*: These applications combine the Service Oriented Applications and the Data Intensive Applications.

Developers of Information system applications are concerned with the data flow (for browsing and retrieving data) and control flow (for the different phases involved in the manipulation of the data).

Electronical book stores or online banking are example of this application type.

4. SOFTWARE ARCHITECTURE

Software architecture determines how system components are identified and the interaction between them and the interface protocols for communication.

4.1. SOFTWARE LAYERS

Software is structured as layers in a single computer or as services offered and requested between processes located in the same or different computers. This process and service-oriented view can be expressed in term of service layers.

4.2. SYSTEM ARCHITECTURES

The division of responsibilities between system components and the placement of the components on computers in the network is perhaps the most evident aspect of distributed system design. The main types of architectural models are the *Client-server model* and *Services provided by multiple servers*.

The **Client-server model** is historically the most important architectural model and remains the most widely employed. A client process interacts with individual server processes in separate host computers. Servers can be clients of other servers. This model is illustrated in Figure 2.

Services provided by multiple servers A client interacts with an server that provide a service to the client. This server interacts with other servers to provide this service, because the servers may portion the set of objects on which the service is based. This model is illustrated in Figure 3.

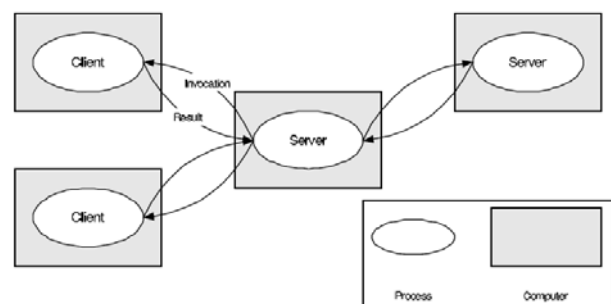


Fig. 2 The client-server model

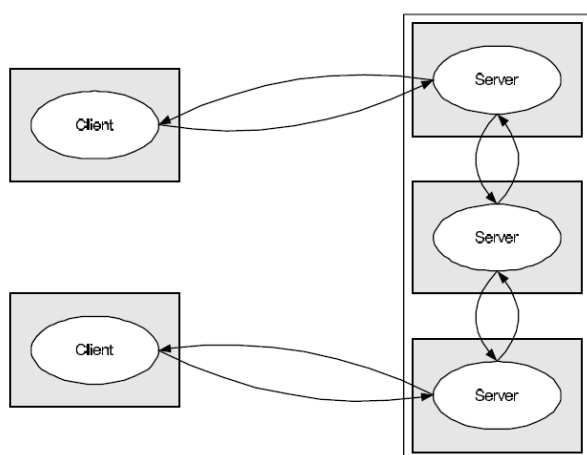


Fig. 3 Services provided by multiple servers

5. SECURITY IN WEBLOGIC SERVER

The Web Application can be securized by using authentication, by restricting access to certain resources in the Web Application, or by using security calls in the servlet code.

To configure authentication for a Web Application, use the *login-config* element of the *web.xml* deployment descriptor. In this element one can define the security realm containing the user credentials, the method of authentication, and the location of resources for authentication.

By default, WebLogic Server assigns the same cookie name (*JSESSIONID*) to all Web Applications. When one use any type of authentication, all Web Applications that use the same cookie name use a single sign-on for authentication. Once a user is authenticated, that authentication is valid for requests to any Web Application that uses the same cookie name.

For implementation and use of user authentication and authorization, BEA WebLogic Server utilizes the security services

of the SDK version 1.3 for the Java 2 Platform, Enterprise Edition (J2EE).

Like the other J2EE components, the security services are based on standardized, modular components. BEA WebLogic Server implements these Java security service methods according to the standard, and adds extensions that handle many details of application behavior automatically, without requiring additional programming.

6. CONCLUSION

This paper describes the field of Web Applications and presents the needs for an own engineering approach for such systems. This approach, called Web Engineering, should have an associated architecture, which takes into account the special features and needs for the development of Web Applications. A architectural framework for Web Applications using architectural views was presented.

BIBLIOGRAPHY

1. Ratiu, I.G., David, N., *Open Systems Architecture*, The 13th International Conference, Computer Science, Modeling and Simulation, E-learning Technologies and Solutions for the Engineering Domain, Conference Proceedings, Land Forces Academy Publishing House, 2007;
2. Fielding, R.T., Taylor, R.N., *Principled design of the modern web architecture*, ICSE 2000 Limerick Ireland, 2000;
3. Kruchten, P., *The 4+1 view model of architecture*, IEEE Software 12(6), 1995.