

---

## Development of ISIRS Web Application Using Object Behavioral Modeling

Haider M. Abdul-Nabi

*Dept. of Computer Science, College of Science, University of Basrah, Basrah ,Iraq*

---

### Abstract:

The growth of the Internet and the World Wide Web (WWW) has resulted in a large number of Web Systems and Web Applications. Today many people are relying on Web applications like Internet banking, entertainment and E-commerce applications. They expect these applications to be available, safe, secure and reliable.

In this research, the purpose is to give an understanding of the special characteristics of Web Applications and the Web Engineering process used to develop these applications. We first analyze the current state of the art of Web applications design, stating the need for an approach that clearly separates concerns (conceptual, navigational, interface). Briefly introduce the ISIRS (Instructors and Students Information & Registration System) application as an object behavioral model of Web applications design.

**Keywords:** Web Engineering, Software Engineering, Web Engineering Process, Web application development.

### 1. Introduction:

A Web application is a software system based on technologies and standards of the World Wide Web Consortium (W3C) that provides Web specific resources such as content and services through a user interface, the Web browser [1].

Web Applications are software applications deployed by the World Wide Web. They use a single client-server model, and run in a Web browser on the client computer. Once a new release of a Web Application is installed on the server, this release is available to all users.

This immediate deployment characteristic is probably one of the most powerful characteristics of a Web Application [2].

Web Engineering is the application of systematic, disciplined and quantifiable approaches to development, operation, and maintenance of Web-based applications [3]. It is a response to the early, chaotic development of Web sites and applications as well as recognition of a divide between Web developers and conventional software developers [4, 5].

Building complex Web applications such as E-commerce applications is a time consuming task. We need to understand the underlying domain (objects, behaviors, business rules, etc). We must carefully design the applications' navigational architecture and user interface, if we want them to be usable.

## 2. Basic Concepts

### 2.1. Web Applications

Web Engineering makes it possible to plan and iterate development processes and thus also facilitates the continuous evolution of Web applications. This permits not only cost reduction and risk minimization during development and maintenance, but also an increase in quality, as well as measurement of the quality of the results of each phase [6].

There are different names in use for what here is called a Web Applications. Names in use are Web Sites, Web-based applications and Web Applications [2]. Some authors are also using different names to indicate different types of Web Applications [7]. In this Research the term Web Application is used to represent all types. A definition of the term Web Application is given in [1]:

A Web application is a software system based on technologies and standards of the World Wide Web Consortium (W3C) that provides Web specific resources such as content and services through a user interface, the Web browser.

## **2.2. Categorization of Web Applications**

As a starting point in understanding the problem domains that the Web currently can address, Table (1) presents taxonomy of Web applications [2]. The order of these categories roughly illustrates the evolution of Web applications.

Organizations that started their Web development early may also have followed a similar order in the past. Although, it is possible to start Web development with applications in any category, this table has been useful to explain to organizations with modest presence on the Web how they might improve or benefit from incremental exposure, thus keeping the risks to the minimum [8].

Table (1) categorize of Web Applications

Category	Example
Informational	Online newspapers, product catalogues, newsletters, service manuals, classifieds, e-books
Interactive <ul style="list-style-type: none"><li>• User-provided information</li><li>• Customized access</li></ul>	Registration forms, customized information presentation, games
Transaction	E-shopping, ordering goods and services, banking
Workflow	Planning and scheduling systems, inventory management, status monitoring
Collaborative work Environments	Distributed authoring systems, collaborative design Tools
Online communities, Marketplaces	Chat groups, recommender systems, marketplaces, Auctions
Web Portals	Electronic shopping malls, intermediaries
Web Services	Enterprise applications, information and business intermediaries

Depending on the category of a Web Application, different resources, techniques and methods are required to develop them. Also, the purpose of Web Applications can change with the category [2, 9]. The complexity of each new category is growing. This approach can be extended, and it is possible to link resources such as methods, processes, and best practices to each category.

### 2.3 Characteristics and Complexity of Web Applications

Web applications vary widely: from small-scale, short-lived services to large-scale enterprise applications distributed across the Internet and corporate intranets. Over the years, Web applications have evolved and become more complex.

They range from simple, read-only applications, to full-fledged information systems. This complexity may be in terms of performance (number of hits per second), for example the Slashdot site [10], the Olympics sites receiving hundreds of thousands of hits per minute [11], or in terms of dynamic nature of information, the use of multimedia or in other ways. They may provide vast, dynamic information in multiple media formats (graphics, images and video) or may be relatively simple. Nevertheless, they all demand balance between information content, aesthetics and performance. Table (2), below, brings out the characteristics of early, simple Web-based systems and current, advanced Web-based systems [12].

**Table (2): Characteristics of Simple and Advanced Web Applications**

<b>Simple Web-based Systems</b>	<b>Advanced Web-Based Systems</b>
Primarily textual information in non-core Applications.	Dynamic Web pages because information changes with time and user's Needs.
Information content fairly static	Large volume of information
Simple navigation	Difficult to navigate and find information
Infrequent access or limited usefulness	Integrated with database and other planning, scheduling and tracking systems
Limited interactivity and functionality	
Stand alone systems	Deployed in mission-critical applications
High performance not a major requirement	Prepared for seamless evolution
	High performance and continuous availability is a necessity

Developed by a single individual or by a very small team	May require a larger development team with expertise in diverse areas
Security requirements minimal (because of mainly one-way flow of information)	Calls for risk or security assessment and Management
Easy to create	Needs configuration control and Management
	Necessitates project plan and management
	Requires a sound development process and Methodology
Feedback from users either unnecessary or not sought	User satisfaction vital
Web site mainly as an .identity. for the current clientele, and not as a medium for communication	Web site/application as the main communication medium between the organization and users

#### 2.4 The Differences and Similarities between Web Applications and Conventional Software:

There are many differences between web applications and conventional software, as shown in Table (3) that summarizes the

experts' findings with a few additional, distinctive characteristics. It is worth noting that this enumeration is based on the experiences of Web developers that the experts had consulted [13, 14, 15, 16]:

**Table (3) The Differences between Web Applications and Conventional Software**

Compressed development schedules	Constant evolution with shortened revision cycles
"Content is king", i.e. it is integrated inextricably with procedural processing	Insufficient requirement specifications
Small teams working to very short schedules	Emerging technologies/methodologies
Lack of accepted testing processes	User satisfaction and the threat from one's competition
Minimal management support	Criticality of performance
Evolving standards to which Web applications should or must comply, depending on the specific circumstances (for example accessibility standards for government sites or IEEE or W3C standards for technological reasons).	Understanding of additional disciplines required for Web applications, such as hypertext, graphic design, information architecture
security considerations	Legal, social and ethical issues
Variety of backgrounds of developers	Rapidly evolving implementation environment, encompassing various hardware platforms

While there are many differences between Web development and Software development, as discussed above, there are also similarities between them. These include:

- Need for methodologies,
- Requirements elicitation,
- Programming,
- Testing

Maintenance of those parts that deal with programming and functionalities.

### 3. Web Application Architectures

In the previous section we have given an indicative overview over references of the state-of-art in web engineering. In this section we will focus on how we can use the existing knowledge in Web engineering. As Web projects often start with a couple of HTML pages and continuously grow to large Web applications.

A 2-layer or (tier) architecture also called client/server architecture uses a Web server to provide services to a client Figure (1) [17].

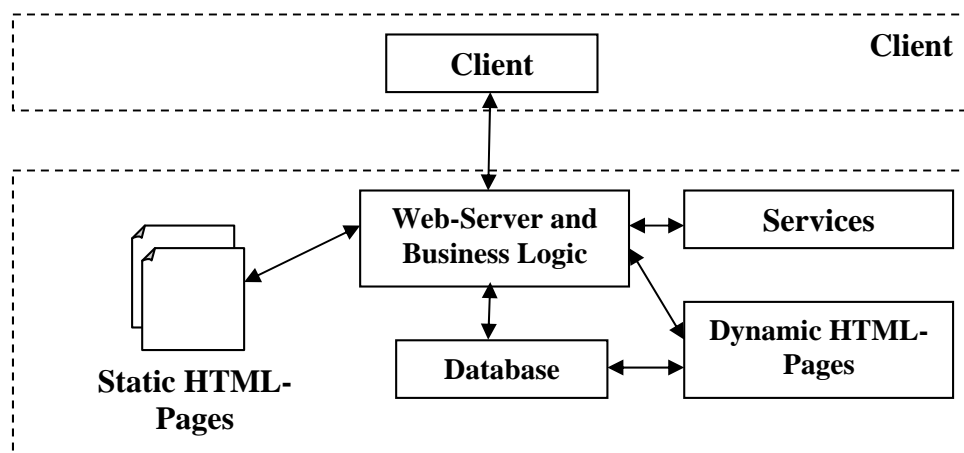


Figure (1) - A 2-Layer Architecture for Web Applications

The 2-layer architecture can take different forms within the environment of Web applications. A client request can point directly to static HTML pages, without requiring any processing logic on the server layer, or it can access a database via the application logic on the Web server (e.g., in the form of CGI scripts). Dynamic HTML pages include script instructions directly in the HTML code, e.g., when SSI (Server-Side Include) is used, and they are interpreted either by databases with HTML functionalities or by a Web server. The application logic, or dynamic HTML pages, can use services (e.g., user identification or data encryption) when the HTML response is generated.

#### 3.1 Application Behavior

The core of every software application (be it Web-based or not) is its domain or conceptual model. It must reflect which objects the application deals with, their relationships and behaviors [18]. In SIS the conceptual model is specified using the UML notation.

The distinctive aspect of Web applications is their navigational architecture: they are basically database applications. We must define which objects (nodes) the user will perceive and how he will traverse the hyperspace (links, indexes, etc). Nodes and Links are defined according the user profile and the task he must perform.

Consider the domain of ISIRS (Instructors and Students Information & Registration System) (i.e., web-based applications that help manage instructor's and student's information in the process of receiving and evaluating their grades for a college of science). A possible conceptual & navigational model for this domain is given in Figure (2). Notice that this model allows reviewers to express their evaluation, marks and degrees in the study year.

For the purposes of supporting the evaluation process, we have defined a "view" over this schema, where Nodes and Links are observers of conceptual classes as shown in Figure (2). Each Node class can be defined by combining attributes from different conceptual classes, while links indicate navigation paths and reflect conceptual relationships. For example, "Student" has incorporated "Stage" & "CourseCode" as an attribute; similarly, "Instructor" has also incorporated "Specialist" & "CourseCode" as an attribute.

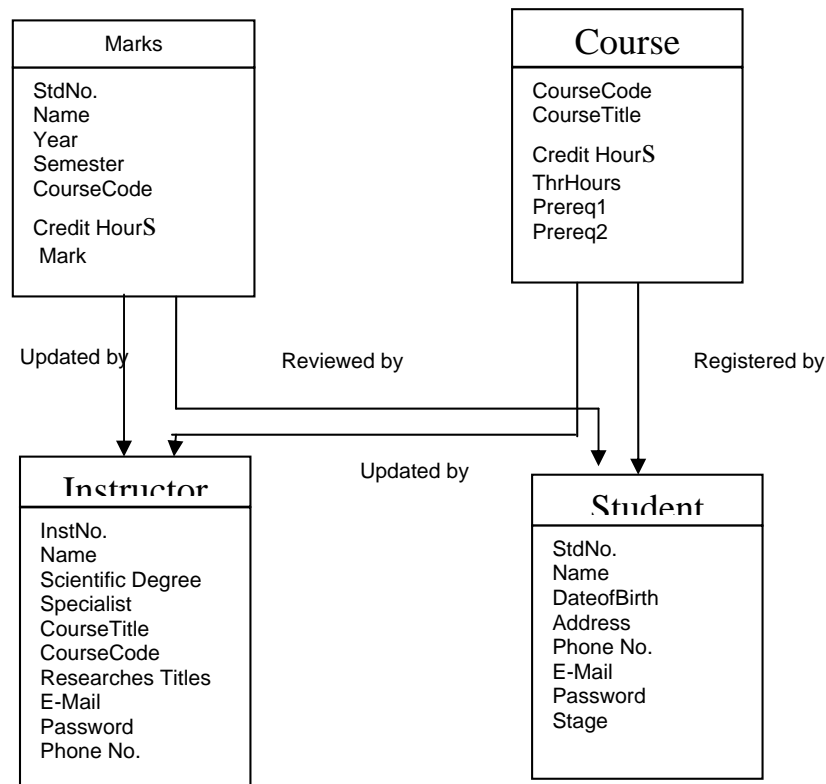


Figure (2) – Conceptual & Navigational Model for the "ISIRS "

---

### 3.2 Object Behavioral Model

A behavioral model shows the interactions between objects to produce some particular system behavior that is specified as (us-cases) which are sequence diagrams (or collaboration diagrams) in the UML used to model the interactions between objects.

Use – cases, also, are a scenario based technique in the UML which identify the actors in an interaction and which describe the interaction itself. Scenarios are real-life examples of how a system can be used as shown in figure (3) student use-case model and figure (4) instructor use-case model.

### 3.3 Execution of the Application

The application for ISIRS (Instructors and Students Information & Registration System) web-site involves three steps. First, the web-server access logs to identify the user sessions by both instructors and students. Next, the ISIRS is used to register courses by student.

Finally, the instructors logs in to modify the students marks. As shown in figure (3) student use-case model and figure (4) instructor use-case model.

1. Students follow the following steps. The student access to the application to display the semester course table, and then log in by his user name and password to his account.
2. After that the student can be select, add, remove the selected course item and there are conditions must be satisfied to verify his selection. Then the registration process is complete.
3. The students can also enter to the application to view their marks.

In turn the instructor log in the application to add, delete, modify the course item, check the student registration process, and modify the students marks.

In the end of semester the student log in to the application by his user name and password to check his final semester degrees.

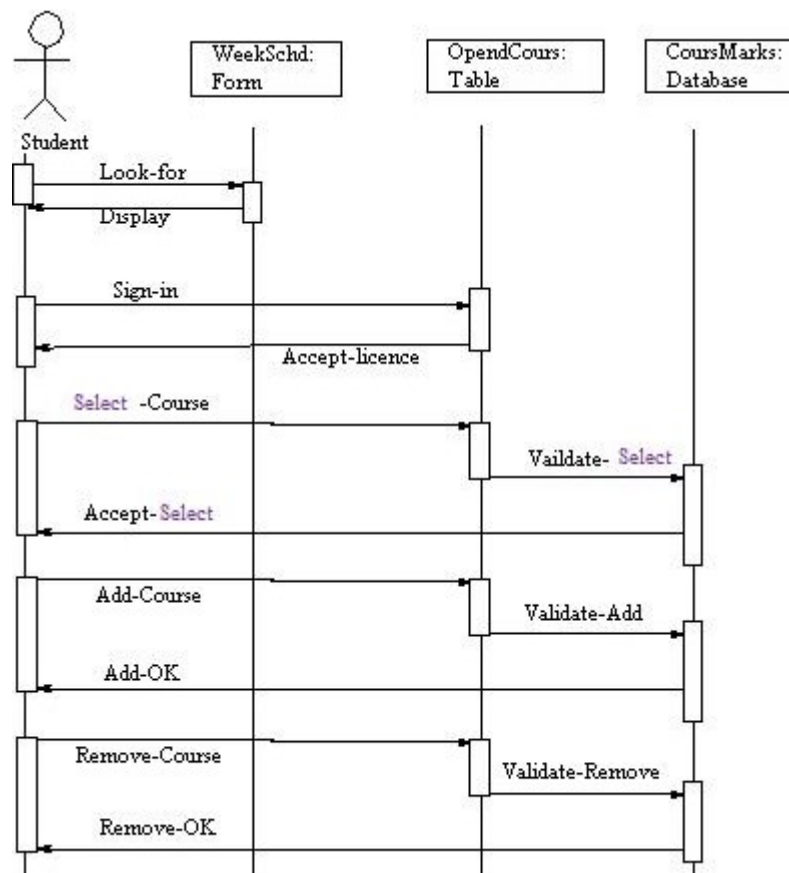


Figure (3) – Student use-case model

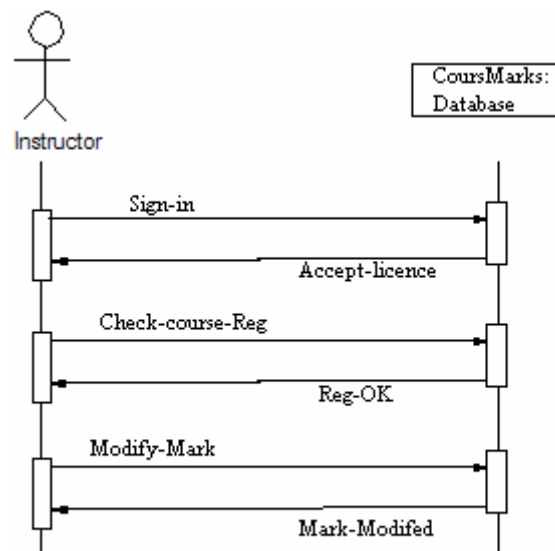


Figure (4) – Instructor use-case model



## Conclusions:

Web Engineering deals with the process of developing, deploying and maintaining Web applications. The main themes of Web Engineering encompass how to successfully manage the diversity and complexity of Web applications development, and, hence, to avoid potential failures that may have serious implications. It is a pro-active approach and at this stages a collection of a body of work. The need for Web Engineering is strong. The task before the Web developers and researchers is to create a robust and tested body of work that can be recommended to suit the specifics of Web applications and environments.

Our application for ISIRS (Instructors and Students Information & Registration System) web-site involves three steps. First, the web-server access logs to identify the user sessions by both instructors and students. Next, the ISIRS is used to register courses by student. Finally, the instructors log in to modify the students marks.

Our case study using ISIRS (Instructors and Students Information & Registration System) web site indicates that, indeed, interesting and useful trends can be discovered in the navigation behavior of the web-site users. Our work is still in a preliminary phase and initial results shows that benefits of this application in tracing the student's information and marks.

## References:

1. Gerti Kappel, Birgit Proll, Siegfried Reich, Werner Retschitzegger, "Web Engineering: The Discipline of Systematic Development of Web Applications", John Wiley & Sons Ltd. 2006.
2. Y. Deshpande, S. Murugesan, A. Ginige, S. Hansen, D. Schwabe, M. Gaedke, and B. White. "Web engineering". Journal of Web Engineering, 1(1):3–17, October 2002.
3. Murugesan, S., Deshpande, Y., Hansen, S. and Ginige, A. (1999) "Web Engineering: A New Discipline for Development of Web-based Systems", Proceedings of the First ICSE Workshop on Y. Deshpande, S. Murugesan, A. Ginige, S. Hansen, D. Schwabe, M. Gaedke, B. White 15 Web Engineering, International Conference on Software Engineering, Los Angeles, May 1999.  
<http://aeims.uws.edu.au/WebEhome/ICSE99-WebE-Proc/San.doc>
4. Pressman, R.S. (1998) "Can Internet-Based Applications Be Engineered?" IEEE Software, September/October 1998
5. Deshpande, Y., Hansen, S. and Murugesan, S. (1999) "Web Engineering: Beyond CS, IS and SE. An Evolutionary and Non-Engineering View", online Proceedings of the First Workshop on Web Engineering, <http://fistserv.macarthur.uws.edu.au/san/icse99-WebE/ICSEWebE-program.PDF>.
6. Ginige, A., Murugesan, S., "The Essence of Web Engineering, Special Issue on Web Engineering", IEEE Multimedia, 8 (2), April–June, 2001b, pp. 22–25.
7. J. Conallen. "Building Web Applications with UML", Second Edition. Addison-Wesley, 2003.
8. Deshpande, Y., Chandrarathna, A. and Ginige, A. (2002) Web Site Auditing - First Step Towards Reengineering, Proc.of the Fourteenth International Conference on Software Engineering and Knowledge Engineering, Ischia, Italy, 15-19 July 2002, pp 731-737.
9. B. Ramesh, J. Pries-Heje, and R. Baskerville. "Internet software engineering: A different class of processes". Annals of Software Engineering, 14(1-4):169–195, 2002.

10. <http://slashdot.org/article.pl?sid=01/09/13/154222&mode=thread&tid=124>.
11. Powell, T.A. (1998) "Web Site Engineering", Prentice-Hall, Upper Saddle River, NJ.
12. Ginige, A. and Murugesan, S. (2001) "Web Engineering: An Introduction", *IEEE Multimedia*, Special issues on Web Engineering, vol 8, no 1, pp 14-18.
13. Pressman, R.S. (2001) "Web Engineering: An Adult's Guide to Developing Internet-Based Applications", *Cutter IT Journal*, vol 14, no. 7, pp 2-5.
14. Glass, R. (2001) Who's Right in the Web Development Debate?, *Cutter IT Journal*, vol 14, no. 7, pp 6-10.
15. Deshpande, Y., Hansen, S. (2001) Web Engineering: Creating a Discipline among Disciplines, *IEEE Multimedia*, Special issues on Web Engineering, vol 8, no 2, pp 82-87.
16. Holzschlag, M.E. (2001), "How Specialization Limited the Web", *WebTechniques*, Sept 2001. <http://www.webtechniques.com/archives/2001/09/desi/>.
17. Anastopoulos, M., Romberg, T., "Referenzarchitekturen für Web-Applikationen, Projekt Application2Web", *Forschungszentrum Informatik (FZI)*, December, 2001, <http://app2web.fzi.de/>, in German.
18. UML Document Set. Version 1.013 January, 1997, Rational, 1997. (Available at <http://www.rational.com/uml/references/index.html>).

### تطوير تطبيق الويب ISIRS باستخدام النمذجة السلوكية للكائنات

حيدر محمد عبد النبي

قسم علوم الحاسبات، كلية العلوم، جامعة البصرة، البصرة، العراق.

#### المستخلص:

النمو المتزايد للانترنت ومواقع الويب (WWW) نتج عنه عدد هائل من نظم الويب وتطبيقات الويب. في هذه الايام معظم الناس يعتمدون على تطبيقات الويب مثل الخدمات المصرفية عن طريق الانترنت، والترفيه، وكذلك تطبيقات التجارة الالكترونية. كما يتوقع معظم الناس أن تكون تلك التطبيقات متوفرة، آمنة، محمية، ويعول عليها. في هذا البحث تم تسليط الضوء على إدراك الميزات الخاصة لتطبيقات الويب وعمليات هندسة الويب المستخدمة لتطوير تلك التطبيقات. بدأنا بتحليل الحالة الآنية لفن تصميم تطبيقات الويب ومن ثم تحديد الحاجة لطريقة تفصل بوضوح الاعتبارات (المفهومية، الاستكشافية، والبيئية). ويقدم بأختصار (ISIRS) (نظام معلومات وتسجيل الطالب المدرس) كنموذج كائني سلوكي لتصميم تطبيقات الويب.

**الكلمات المفتاحية:** هندسة الويب، هندسة البرمجيات، عملية هندسة الويب، تطوير تطبيقات الويب.

This document was created with Win2PDF available at <http://www.daneprairie.com>.  
The unregistered version of Win2PDF is for evaluation or non-commercial use only.