

An Investigation of Design Methodologies for Internet Business Application Development

Track Title: Technology and Information Management

Maneli Rodriguez-Medina, MyCity.Com, Miami, FL.

Ronald Giachetti, Department of Industrial & Systems Engineering, Florida International University, Miami, FL

Abstract

Web companies must quickly design, develop, and bring to market new products. This is especially necessary with the recent shift in venture capital firm's attitudes from ambivalence toward profits to wanting to see the company is on a path towards profitability. Because this business is relatively new, it lacks a methodology or standardized processes for developing new web applications. Various systems analysis and design methods are reviewed in this paper. An assessment of the unique requirements for designing and constructing web applications are made with the purpose of establishing a new methodology for web application design. This approach is an adaptation of object-oriented analysis and design with extensions to the unified modeling language.

Introduction

Web-sites have evolved from static web-sites, well known as "brochure ware", where the main purpose was posting information for marketing purposes to highly interactive or dynamic web-sites for e-commerce. Here we define a *web application* as a web site where user input via navigation and data entry changes the state of the business. Thus, a web-site that allows visitors to order a product is a web application since the order represents a change in the business state (an outstanding order was input to the system), whereas a web-site that only lets users view information is not a web application. The characteristics of web applications have inspired different authors (e.g. Powell 2000) to propose that web applications should be treated as software applications. However, there are some significant differences between web applications and software. First, partitioning functionality onto the web client/server architecture is important for performance reasons. Certain technologies are inherently designed for either client-side only (e.g. JavaScript) or server-side only (e.g. Cold Fusion). Proper architectural design and partitioning of functionality in the architecture is necessary to ensure scalable web applications. Second, while we wish to incorporate functionality onto the web application the user interface remains paramount. Web applications are designed for a broad audience that has different experience levels, browser technologies, and connection speeds which all must be considered in the user interface design. Moreover, the user interface for an e-commerce application is akin to the storefront for a tradition store and must convey the corporate image and branding in order to compel users to enter, otherwise the competition is a click away. Few software projects internal to firms have such an importance attached to the user interface. Third,

hypermedia design concepts remain important due to the intermingling of content and function in web applications. Therefore, hypermedia design concepts of (1) organizing a large body of information into numerous fragments (pages), (2) pages relate to each other, and (3) user needs to see only a small fragment of the information at any one time, are all important in web application design. Finally, web design projects will usually include a large number of people with different skills and expertise concerned with different aspects of the project. Programmers, artists, content designers and authors may be involved in a web project. Consequently, web application design has the characteristics of a software design project with additional requirements unique to the medium of the world wide web.

Based on our realization of the differences between software design and web application design we believe the best approach is to adapt software design methods and tools to meet the specific requirements of web applications. In this article we first trace the evolution of web technologies and then review various systems analysis and design methods. An assessment of the design process is used to make suggestions for improving the performance of the design process through method changes and modeling tool extensions.

Evolution of web technology and development

Internet Business has changed dramatically in its still short life. At the beginning of 1990 few companies had realized the importance of being part of the Internet technology revolution. The use that they were giving to the web was for marketing purposes only; whereas today the focus has shifted to the greater functionality found in web applications. Functionality is accomplished through dynamic web applications, where customers and business partners are connected to their vendors' database and perform different activities that enhance the business operations in real time. A study performed by The Gartner group in 1999, states that the web has evolved in four stages. The first stage is the *Basic Presence* in which companies deploy information about their products and services in the form of web-site. Pages are static ads and are limited in number. The second stage is named *Prospecting*, in which companies add many more pages of information about the company and its products. The most important aspect in this kind of web-site is placed on the information provided and the aesthetic presentation of the information; technically translated as content and design. As Hiuzingh (2000) states in his study content refers to the information features or services that are offered by the web-site, while design is the way the content is made available for web visitors. Visitors can interact with the web-site to request additional information by submitting forms to the site and can conduct key word searches of web-site information. The third and more advance stage is *Business Integration*. It supports eCommerce transactions. In this stage the web-site has full-text search capability throughout and provides services to customers, including account and delivery information, support and community-building features such as online chat, hosted web pages and other activities. The fourth and last common stage is *Business Transformation* where the web-site changes nearly every aspect of the company's business processes and become integral to the business. Web-sites at this stage of development provide electronic document interchange with suppliers, vendor integration services, and content personalization services for site visitors. It is the later stage that we refer to as web applications.

Software Design Methods

Waterfall Model – Describes the lifetime of a software project as serial stages. Defined by planning phase, design phase, implementation phase, and testing phase and maintenance phase [Kendall 1992]. The advantages of the waterfall method or traditional life cycle methodology are it forces developers to plan everything up front, is easy to understand, and the steps are easily monitored, and can serve as project milestones. One drawback often cited is that it is very time consuming. The difficulty in returning to reconsider earlier decisions means that each phase requires very careful completion, since mistakes that are made can be costly and difficult to correct.

Modified Waterfall – it is a modification of the traditional waterfall, trying to add more concurrent activities in this sequential process. This approach suggests that the planning stages of the project iterate a few times to reduce risk by trying to throw out the not critical aspects of the project [Powell, 1998].

Rapid Application Development (RAD)- approach for building computer systems which combines Computer Assisted Software Engineering (CASE) tools and technique, user driven prototyping and stringent project delivery time limits into a potent, tested, reliable formula for topnotch quality and productivity. RAD seeks to improve the quality of finished systems by working directly with end-users while reducing the time it takes to build them by iterative design/build/test.

Spiral Model – The basic idea of the spiral model is to incrementally develop the site by breaking development up into many subprojects. One of the main points of the process is to attempt to reduce or at least manage risks by focusing on the most critical aspects of the system first. Powell (1998) argues that this model best addresses the needs of web application development but the method can be considered complicated and difficult to manage.

Object Oriented Analysis and Design – are based on object oriented programming concepts, can help respond to organizational demands for new systems that must undergo continuous maintenance, adaptation and redesign [Rumbaugh et al. 1991; Jacobson et al. 1992]. There are many variations of the object-oriented analysis and design method. This method is very flexible, and support the reuse of object, reducing the time and cost. It is helpful basically because allows the simultaneous work (concurrent jobs). Once the project is separated in objects, each programmer can work individually; at the end the work is merged resulting the final project [Allen and Frost, 1998].

Joint Application Development (JAD) – is a management process, which specifies how systems analysts and designers can work effectively with users to develop solutions. JAD, developed by IBM is an alternative approach to interviewing users individually. The motivation for using JAD is to cut the time (and hence the cost) required by personal interviews, to improve the quality of the results of information requirements assessment, and to create more user identification with the new information systems as a result of the participative process. JAD has usually been employed as a technique that allows system analyst to accomplish requirements analysis and design the user interface jointly with the users in a group setting.

The Relationship Management Methodology (RMM) - [Isakowitz et al. 1995] uses the relational database model and extends the notation by access structures to achieve a navigational model of

the resulting hypermedia application. It is a well known approach for hypermedia design, of which web applications are related.

Web Application Development Tools

Most methodologies are associated with tools. Table 1 summarizes four broad classes of tools commercially available for web application design. WYSSYG commercial web development environments, such as Macromedia Dreamweaver or NetObject's Fusion, concentrate overwhelmingly on the graphics design. Little support is provided for design other than navigation. Tag-based design environments such as Allaire's Homesite provide good support for rapid development of HTML and CFML code but again do not support any modeling for analysis and design. The leading CASE tools for software engineering do not yet support web application development. Largely, the reason for a lack of commercial tools for supporting web application design is the fast emergence of the technologies and an early emphasis on graphics and content design. Early corporate web sites were for marketing and now as corporations increasingly incorporate e-commerce and other business functionality the need for modeling grows. Ironically, the ease and success of the commercial web development tools have the unfortunate side effect of leading people into believing that web page design is a simple task. Yet except for the most trivial web sites adherence to a design methodology is necessary to tackle the complexity of the web application, especially with regard to e-commerce.

Table 1. HTML Development Tools

Tool	Pros	Cons
Hand coding	Provides most controls Able to use latest tags	Slow process Prone to making errors Requires knowledge of HTML Difficult to pre-visualize a page
Tag Editors	Faster than by hand Enable the insertion of tags with appropriate syntax Enable editing of attributes and links	Requires knowledge of HTML Difficult to pre-visualize a page Syntax checkers might feel constrictive.
HTML WYSWYG	Enables visual design of pages Easy to use Do not requires HTML knowledge	Provides de least amount of control over the code Contains odd nuances Difficult to edit by hand Built-in display mode for pages that do not match pages as actually rendered by browsers
Translators	Able to convert existing documents quickly	May still require hand or editor cleanup.

In software development the Unified Modeling Language (UML) is an attempt to provide a standardized notation for describing object oriented models [Rumbaugh et al. 1999]. Many vendors provide CASE tools for the UML. The UML defines a collection of diagrams, the associated syntax, and semantics for defining an object-oriented system. These diagrams are enumerated in 9 categories described as: *Class diagram*-shows the entities in a system or a domain and how those entities relate to one another. Each class is represented as a named rectangle. *Object diagram*-shows instances of the classes and their interrelationships. Each object is represented as a named rectangle. *Use Case diagram*-shows system usage. Each use case appear as an ellipse, and each actor as a stick figure. *State diagram*-captures the state of an object during a specific time period. A state is represented as a rounded rectangle, a transition between states as a line connecting those states. *Sequence diagram*-visualizes how the objects in a system interact with one another over time. The objects are laid out across the top, and time proceeds from the top of the diagram to the bottom. Arrows denote messages that go from object to object. *Activity diagram*-show the steps and decision points that occur within the behavior of an object, or within a business process. Each step is a rounded rectangle (more oval shaped than the representation of a state) and each decision point is a diamond. *Collaboration diagram*-is another way of visualizing how objects work together over time. Objects may be anywhere in the diagram. Messages from one object to another appear as lines connecting the objects. Each line is numbered according to its placement in the sequence of messages and shows information that pertains to the nature of the message. *Component diagram*-models the software components of a system. Each component appears as a rectangle with two smaller overlaid rectangles on its left border. *Deployment diagram*-represents the physical architecture of a computer-based system. It can show each computer and device in the system and the components that reside in each computer. The computer or node is represented as a cube with the individual components within.

Comparison of software design methods and web application design requirements

Web application design requires both information or content design and functionality design. The user interface plays a prominent role in the success of the web application. Clearly application of hypermedia design concepts for information delivery, navigation, and inclusion of cognitive aspects of the user interface would greatly benefit the design of web applications. In both software design and web application design product time to market and product quality issues are mentioned by Sawyer [1998] as critical in the success of the final product. Managing the complexity of the web application is another development requirement. Web applications are complex due to the various and disparate technologies used to build them and especially with regard to the distributed nature of the technologies in the client/server architecture. Moreover, the web application development methodology needs to provide a means for continuous improvement or continuous changes. Even with adequate decomposition of the various functions of a site, the technology used to implement sites is changing very rapidly, making it difficult to maintain web applications.

The strong pressures to reduce time-to-market delivery of software lead to the adoption of RAD and JAD techniques instead of more traditional waterfall type development models. JAD however is not amenable to web application design in many instances because the system user is not readily available. Thus, JAD is best when the end user is internal to the company. RAD can be applied to web application development. RAD is strongly associated with RAD development environments. Some RAD development environments such as Allaire's ColdFusion are

available. However, RAD alone does not address all of the web application development needs. For example, RAD does not specifically address performance issues, managing of complexity, or providing for continuous improvement. RAD is therefore best used in conjunction with other methodologies. Object-oriented analysis and design does address the later three requirements. Dividing the project in components and classes allows many developers to be involved with the project, helping to minimize the time of design and implementation. Also the reusability is well accomplish in this approach due to the classes and objects segregation. The RAD methodology is implicitly utilized in the development phase, but not implemented as a methodology per se. In the object-oriented design the UML is used to describe and communication the design between the development team [Rumbaugh et al. 1999].

Web Application Development Methodology and Supporting Tools

The UML defines the syntax and semantics of modeling constructs but does not propose any design process. Our design method utilizes three modeling tools: use cases, class diagrams, and sequence diagrams. We adhere to the following object-oriented design method:

1. Requirements: In these phases an analyst, architect and designer are in constant communication to define the product to be designed. The functional requirements of the web application are gathered through use case modeling. A use case is a scenario describing the interaction between a user and the system.
2. Analysis: In the analysis phase we analyze each use case and determine what classes are required, the class structure, and necessary interaction between objects in order to complete the functions. The analysis models are logical models in that they do not convey implementation details. The UML class diagrams and sequence diagrams with analysis objects are utilized in this phase and our extension to the UML of navigation structure.
3. Design: Analysis models are refined to incorporate the physical implementation of the system. This includes partitioning of objects to either client or server side technologies, separation of pages, navigation structure, and database implementation issues. The UML class diagrams and sequence diagrams are utilized in this phase.
4. Construction: The design models are mapped into code. CASE tools actively support this phase.
5. Test: The web application is tested for functionality and performance. Additionally, web applications should be extensively user tested for usability and human computer interaction.

We recognize the power of rapid application development techniques and the above phases would be conducted iteratively throughout the web application development project.

We apply use cases to capture functional requirements [Jacobson, et al. 1992]. A use case shows the actors and the functions they perform with the web application. During the analysis phase we identify the classes from the use case descriptions. We then capture the behavior of the classes by modeling object collaboration in a sequence diagram. During design we revise the analysis diagrams to partition among architectural components, determine responsibilities of the classes, and to incorporate interface objects, control objects, and model objects. It is in the design phase we apply our stereotypes object descriptions for modeling the web application.

Until the design phase we can implement the logical system description using any technology; in design we make these design decisions.

We extend the UML based on the hypermedia design concepts [Isakowitz et al., 1995] in order to capture necessary navigation. Additionally, we utilize the web application extensions (WAE) to the UML proposed by Conallen

Conclusions

The fact that software is designed in distinct phases from gathering user requirements to coding a tangible software product strongly relates it to web application design. Current web applications utilize object-oriented and a variety of other technologies which are well suited to be modeled with the UML. The UML provides a graphical model of the system that facilitates communication among the development team, users, and managers. To be effective, the UML needs to be used in conjunction with an object-oriented analysis and design method. We analyzed different development methodologies and argue that the object-oriented analysis and design methodology merged with RAD techniques and extended for the specific requirements of web applications is the most suitable. This approach seems to be a good starting to deliver a standardize methodology to the needful field of web application design.

References

- Conallen, J., Modeling Web Application Architectures with UML, *Communications of the ACM*, 42(1), 1999, 63-70.
- Huizingh, E. The content and design of web-sites: an empirical study, *Engineering*, (2000).
- Isakowitz, T., Stohr, E., and Balasubramanian, P., RMM: A Methodology for Structured Hypermedia Design, *Communications of the ACM* 38(8) (1995) 34-44.
- Jacobson, I., Christerson, M., Jonsson, P., and Overgaard, G., *Object-oriented Software Engineering: A Use Case Driven Approach* (Harlow, England, Addison-Wesley 1992).
- Kendall, K.E., and Kendall, J.E., *Systems analysis and Design*, (1992).
- Allen, P. and Frost, S. *Component- Based Development for Enterprise Systems*, (1998).
- Rational Software Corporation, *Building Web Solutions with the Rational Unified Process: Unifying the Creative Design Process and the Software Engineering Process*.
- Rumaugh, J., et al., *Object-Oriented Modeling and Design*, (Prentice Hall International, NJ, USA (1991).
- Rumbaugh, J., Jacobson, I., and Booch, G., *The Unified Modeling Language Reference Manual* (Addison-Wesley, Reading MA, 1999).
- Sawyer, S. and Guinan, P.J. (1998), Software development: Processes and performance, *systems, Information System Management Magazine*, Fall 1999.
- Powell, T.A., Jones, D.A., and Cutts, D.C., *Web Site Design* (1998).