A Framework for Web-Enabled Large-Scale Decision Support System Engineering

Nabil Arman, Palestine Polytechnic University, Hebron, Palestine

Abstract

Developing a large-scale information system (IS) can be a complicated and a challenging exercise. Combined with the complexity of developing a Decision Support System (DSS) and the complexity of developing Web-enabled systems, developing a Web-enabled large-scale DSS designed to serve multiple managers in different business functions who might be distributed over a large area can be a more challenging task than that of developing the much more common two-tiered client/server DSS that have evolved over the past few years. In this paper, it is demonstrated that extra efforts and a more systematic approach in Web-enabled large-scale DSS development are worthwhile, since such a system should incorporate and benefit from the recently developed technological techniques, including information distribution and data exchange schemes. An attempt is made to identify those characteristics of DSS that require different treatment and attention and some issues, which can have greater effect on developing Web-enabled large-scale DSS, are presented..*

1. Introduction

A DSS is a computer-based information system designed to help managers solve problems in semi-structured decision-making areas [1]. Successful DSS applications have addressed problems and decisions in a broad range of managerial environments. By definition, semi-structured decision-making environments are those not well enough understood to permit complete analytical description. Structured processes refer to routine and repetitive problems for which standard solutions exist, whereas unstructured processes are fuzzy and complex problems for which there is no cut-and-dried solution [2].

Planning and analysis are critical tasks in the development of large and complex information systems, in general, and Web-enabled large-scale DSS, in particular, especially those designed to support and serve multiple managers in different locations. Differences between two-tiered client/server DSS and Web-enabled DSS oblige developers to address the main issues affecting Web-enabled systems in general. In this paper, these differences are approached from an applications perspective. Developing a Web-enabled large-scale DSS designed to serve multiple managers in different business functions who might be distributed over a large area can be a challenging task than that of developing the common two-tiered client/server DSS that have evolved over the past few years. In this paper, it is demonstrated that extra efforts and a more systematic approach in Web-enabled large-scale DSS development are worthwhile, since such a system should incorporate and benefit from the recently developed technological techniques

2. Development of Web-Enabled Large-Scale DSS

As mentioned in the introduction, the development of a Web-enabled large-scale DSS is different from the development of a traditional two-tiered client/server system. In the following sections, the main issues that are somehow different are highlighted.

2.1 Planning and Analysis Stages of Web-Enabled Large-Scale DSS Development

Although the following discussion applies to DSS development, in general, it is more appropriate to Web-enabled large-scale DSS. The reason is that two-tiered and Small-Scale DSS development does not generally have the necessary resources, in terms of time and budget that are a must for large-scale systems. The planning stage should proceed by conducting a decision support analysis, which includes interviews, decision analysis, data analysis, and technical analysis. More attention and focus should be given to the consolidation effort among all these activities. Then, DSS software evaluation and selection, which includes a list of candidate vendors and a credible feature analysis, should be conducted. Again, more attention should be on the technical issues related to Web support. This should include a concrete evaluation of major Web Servers and their support for the major technologies, including easy access to major database servers, XML support, Server-Side Java Support …etc. Since a set of databases, possibly from different vendors, may be used in a large-scale system, the issues of integration and data exchange are of great importance.

Requirements analysis is costly and may be hard to justify for DSS in general. Combined with the extra costs that are associated with Web technology is a real challenge. However, in this era of so many mergers between and among different companies and enterprises, requirements analysis should be performed with extra caution in a Web-enabled large-scale DSS. It is known that the decision support needs of DSS users change frequently, and taking into account that so many users/managers from a large enterprise will be using the system, a further step is needed to consolidate all these requirements into a unified set of requirements that will meet all the needs.

These factors and others have led many authors to argue for an evolutionary approach to DSS development using prototypes and rapid development tools such as fourth generation languages. However, many of these tools are not Web-enabled and using prototypes developed using them might not give mangers the right impression about the system.

* DSS is an abbreviation of singular and plural form (determined from the context)

2.2 Design and Implementation of Web-Enabled Large-Scale DSS

At this point, a proposed Web-enabled large-scale DSS architecture that supports the suggested framework is laid out. The architecture consists of a number of layers where each layer is responsible for a specific set of tasks. The architecture allows for the needs of the following categories as pointed out by Mallach [3]:

All levels of management
All decisions (unstructured, semi-structured, and structured)
All major functional and geographic divisions of the organization
This architecture should reflect these elements:

(a) Data Services

Data services are essential for the management of information bases. The use of "information bases" rather than databases (whenever possible) is needed for Web-enabled large-scale DSS. Information bases are distinguished from transaction-processing databases. Information bases are relatively small, and usually highly aggregated, reflecting the fact that managers usually overestimate the amount of transaction detail required to satisfy their query and analysis needs with an eye toward the future. Information bases emphasize multiple scenarios and alternative views rather than consistency and completeness.

Data services are supposed to be used on top of the database management systems (DBMS) to handle all issues of data extraction necessary for Web-enabled large-scale DSS. These data services are essential to ensure that multiple scenarios and alternative views are handled appropriately.

(b) Model Services

The function of model services is similar to that of data services. The main function of data services is the separation of system users, in terms of independence of the application, from the physical aspects of database structure (generally referred to as program/data independence). Similarly, model services are intended to provide independence between the models that are used in a Web-enabled large-scale DSS and the applications that use them. The primary purpose of model services is to extract and transform the relevant data from the information bases using the data services into information that is useful for decision making.

Model services are supposed to be used on top of model-base management systems (MBMS). The functions of a MBMS [1] include creation, storage, access, and manipulation of models. All issues related to MBMS and their objectives are well-documented in the DSS literature.

(c) Dialog Generation and Management Services

The primary function of dialog generation and management services is to enhance the ability of all managers in all different locations to benefit from the Web-enabled large-scale DSS. These services are supposed to handle the support for Web technology, to benefit greatly from that in terms of the representation of information using hypertext markup language (HTML), and the information exchange using eXtensible Markup Language (XML). HTML provides several different ways for presenting information in tabular forms, graphical forms, iconic forms, frames, and so on. Everybody knows how easy it is to present information in any of these forms. HTML handles text and graphic information that allows managers of a DSS to jump from a given topic/alternative to another, whenever they wish, or to jump to related alternatives as they deem necessary. Thus, HTML provides managers of an open ended and a totally controlled way of viewing information.

It is well-known that HTML documents allows users/managers to retrieve and access information in a nonlinear fashion to the greatest limit possible and to control the level of details needed at each stage of using the system. Combined with the power of XML, which describes a class of data objects called XML documents in terms of providing a de facto standard for data exchange, these technologies will have a great benefit for Web-enabled large-scale DSS. In fact, combined further with eXtensible Style Language (XSL) and cascading style sheets (CSS), HTML and XML provides a great technology for information presentation, where XML handles the information/content and HTML with XSL and CSS handle the information presentation. In fact, the same XML document can be presented in so many different ways using different style sheets to reflect how different managers would like to view information needed in their decision-making practices.

XML provides full support for double-byte Unicode character set, as well as its more compact representations, and thus foreign languages are fully supported. Multi-lingual and Unicode support have been a major concern for E-business applications. However, XML will help greatly in tackling this issue.

Dialog generation and management services are supposed to be used on top of dialog generation and management systems (DGMS). If possible, in a Web-enabled large-scale DSS, the information provided by a DGMS may be used, which might not be in HTML or XML documents, and generate XML documents out of that. In addition, dialog generation and management services would use Web services as provided from different vendors to provide the standard Web servers functionality. The new software development tools like Microsoft .NET Framework

Software Development Kit (SDK), which is designed to help in building powerful applications and services based on .NET Framework technology, simplifies greatly the implementation of the suggested framework [11].

In this framework, users/managers are able to use the system from any location by just having thin client machines capable of running an Internet browser. Any constraints/restrictions relating to security, jobs, or manager levels are handled in the standard way.

At the implementation stage, any constraints relating to hardware and operating systems should be stated. For a large-scale system, high-end database servers, operating systems, Web servers are supposed to be used. If there is a corporate policy to standardize on products from a specific vendor, that should be specified at this stage.

2.3. Web-Enabled Large-Scale DSS Integration

Web-enabled large-scale DSS integration can be divided into two main categories: Technical integration and non-technical integration [4]. In section 2.3.1, the technical integration is discussed, and in section 2.3.2, the non-technical integration is presented.

2.3.1 Web-Enabled Large-Scale DSS Technical Integration

It is essential to address the design of new large-scale DSS and the integration of existing and new systems, along with the capabilities provided by the Web technology. In addition, the operation, maintenance, and management of these systems should be addressed. There are four aspects from which system integration can be viewed [1]:

Integration Technology: This has to address issues related to implementations that support data exchange across different subsystems.

Integration Architecture: This has to address and ensure easy and secure data sharing across different subsystems.

Semantic Integration: This has to address the issue of resolving semantic inconsistencies among different concepts used in the system.

User Integration: This has to address the issue of user interfaces and their adaptability and how user friendly the interfaces are.

There are other aspects of technical integration that should be addressed, including:

a) Inter-operability, which is the capability to organize and transfer information between scientific models and cross-functional components of the integrated system.

b) Scale, which is the ability to transfer modeling output to other scales without losing the validity of the information, is necessary for effective decision making.

c) Communication, which is the exchange of information in a common frame of reference, should also be addressed.

To create a DSS that will cope with a decision making process on local as well as national scales, development of a formal framework for communication is paramount. This framework should be suitable for human as well as automated processing. Given the capabilities currently demonstrated in the use of the Internet and the Web technology, system connectivity is not a major problem.

2.3.2 Web-Enabled Large-Scale DSS Non-Technical Integration

The development of complex Web-enabled large-scale DSS has paralleled the rise in more integrated approaches to management. With traditional single resource management, DSS consisted of models which projected future availability of managed resources. However, although a DSS can be a highly effective tool, these problems cannot be addressed by the application of leading edge technology alone. The final decisions will still have to be made by the managers.

Communication is another factor of the key elements to integration. All parties involved with a Web-enabled large-scale DSS construction must be able to freely communicate ideas and concerns. For example, news groups and discussion forums are becoming so popular and they represent a useful tool for facilitating communication.

3. Web-Enabled Large-Scale DSS Specific Requirements

Web-enabled large-scale DSS development requires some additional requirements. Although, some of these requirements have been mentioned before, they are mentioned again to stress the fact that they are of great importance. These requirements will be discussed in the following sections.

3.1 Use of Information Bases

One of the key elements in effective DSS is a set of well-designed, multi-dimensional data structures that allow alternative views of important business variables. Research confirms that end users perceive the need for such capabilities [5, 6]. Different responsibilities imply different views of strategic and operational information. At the same time, effective communication between managers requires consistent procedures for structuring and aggregating information across several dimensions. Some of the keys to effective design of the multidimensional data structure include easy access, easy restructuring of information and manageable dimensionability.

3.2 Improving Data Legibility in Web-enabled large-scale DSS

Data legibility is knowledge workers' ability to understand fully and immediately the data contained in a decision-support system [7]. This is the single most difficult problem in Web-enabled large-scale DSS design. To be effective, DSS environments must provide readable, clear, and navigable data to the knowledge worker or end user. A Web user interface is very beneficial in this regard. Also, the data must present itself in a fashion consistent with knowledge workers' business models, and must promote swift and simple formulation of business questions consistent with knowledge workers' goals.

3.3 Data Warehouses and Data Marts

As mentioned before, data services are supposed to be a custom-built software layer that includes some of the functionalities and capabilities of data warehouses and data marts. Data warehouses are not cheap and multimillion dollar costs are common. Their design and implementation requires considerable time to create. Being designed for the enterprise, so that everyone has a common data set, they are large and increase in size with time. A scaled-down version of the data warehouse is the data mart. A data mart is a small warehouse designed for department level. It is often a way to gain entry and provide an opportunity to learn. Because of the facts mentioned before, data services might be a more feasible solution than using a data mart or a data warehouse.

3.4 Integrated Computer –Aided DSS Engineering (I-CADE)

Computer-Aided DSS Engineering (CADE) is the automation of the DSS Engineering discipline. When the word “I-CADE” is used, “I” stands for integrated. There are two levels for CADE tools: Lower and Upper. Lower CADE tools are used for programming support only and Upper CADE tools are used for conceptual or logical design. While these differences do exist, the general assumption is that the interest is really in I-CADE where one expects these tools to be for all phases of DSS development. I-CADE should support the generation of interfaces that use HTML and other Web languages to be useful for Web-enabled large-scale DSS.

An ideal CADE should provide complete automated support for the entire project life cycle, beginning with enterprise level analysis and working through to maintenance and requirement. The ideal CADE then becomes the focal point for all work that takes place in DSS engineering, and the work of a DSS developer concentrates on the logical aspects of design. An ideal CADE should allow users to integrate different applications easily, which is of major importance for a Web-enabled large-scale DSS, where integration of different applications, as well as the integration of different components of a Web-enabled large-scale DSS, is a must. If CADE is of some importance to a small-scale DSS, then it is critical for a Web-enabled large-scale DSS, since a larger development effort is expected, and the help of automated tools should be invaluable.

3.5 Reusability in Web-enabled large-scale DSS Development

In its basic form, reusability is a property of a code module such that the module can be used, as is, by several applications. In designing for reuse, the goal is to identify modules for potential reuse. In software development, the two most popular methods of implementing code reuse are program templates and reusable modules. The emphasis here should be on the use of components that are useful for Web applications in general.

3.6 Web-Enabled Large-Scale DSS Development as an Application of Client/Server Architecture

Using client/sever architecture will be of great value for a Web-enabled large-scale DSS. As mentioned before, a Web-enabled large-scale DSS will be used by many users for different business needs. Therefore, it would be necessary to keep all the functionality on a powerful server, with a minimal software to be installed on client machines. In this case, money and effort can be saved. In fact, the driving force behind client/server computing is the fundamental belief that inexpensive PCs attached to servers offer the ultimate in price and performance. For the most part, this is true. Although, there is no universal agreement on the client/server advantages, the most often projected benefits include, but are not limited to, adaptability, reduced operating costs, platform independence, and easier data access [8,9,10].

These benefits are needed for a Web-enabled large-scale DSS (as well as for other large-scale information systems). Though client/server architecture can be very complex, there are, generally speaking, two kinds of client/server infrastructures to choose from: two-tiered and three-tiered (N-tiered, where N>3, is also used, however, three-tiered architecture demonstrates all the benefits of N-tiered architecture). The choice between a two- and three-tiered architecture should be based on the scope and complexity of a project, the time available for completion, and the expected enhancement or obsolescence of the system.

Accessing a Web-enabled large-scale DSS using the Internet is one of the main advantages for using three-tiered architecture. Therefore, one can take benefit of the World Wide Web. For example, a company with many branches distributed across a large area and that has a Web-enabled large-scale DSS gives its manager the benefit of accessing the system using the Internet from any place that has Internet access. In addition, with three-tiered architecture one can take the benefit of the lower

maintenance cost of the system, since the functionality of the systems is centralized at the server. In addition, what a manager needs is just a capable Internet browser and the proper authorization to access the system.

4. Conclusion

Developing a Web-enabled large-scale DSS designed to serve multiple managers in different business functions who might be distributed over a large area can be a challenging task than that of developing the common two-tiered client/server DSS that have evolved over the past few years. In this paper, it is demonstrated that extra efforts and a more systematic approach in Web-enabled large-scale DSS development are worthwhile, since such a system should incorporate and benefit from the recently developed technological techniques. An attempt is made to identify those characteristics of DSS that require different treatment and attention. Then some issues which can have greater effect on developing Web-enabled large-scale DSS are presented. Web-enabled large-scale DSS are still a class of large-scale information systems and that all new approaches and ideas that have been developed and proved to be useful for large-scale information systems are adaptable, in one way or another, to Web-enabled large-scale DSS. Of course, there are specific requirements for Web-enabled large-scale DSS, but the overall architecture is somehow similar to large-scale information systems.

5. References

[1] Sage, A. *Decision Support Systems Engineering*, John Wiley & Sons Inc., 1991.

[2] Turban, E. *Decision Support and Expert Systems*, 3rd Edition, Macmillan Publishing Company, New York, 1993.

[3] Mallach, E. *Understanding Decision Support Systems and Expert Systems*, Irwin/Mc-Graw Hill, 1994.

[4] Bonnel, B., "Integration: The Key to the Successful Development of a Large-Scale Forest Management Decision Support System," *Canadian Forest Service*.

[5] Kerschberg, L., Gomaa, H., Menasce, D., and Yoon, J., "Data and Information Architectures for Large-Scale Distributed Data Intensive Information Systems," *Proc. IEEE Eighth International Conference on Scientific and Statistical Database Management*, Stockholm, Sweden, June, 1996.

[6] Gomaa, H., Menasce, D., and Kerschberg, L., "A Software Architectural Design Method for Large-Scale Distributed Information Systems," *Distributed Systems Engineering Journal*, 1996.

[7] Demarest, M., "Improving Data Legibility in Decision Support Systems," *DBMS Magazine* (7:5), May 1994, p55.

[8] Renaud, P. *Introduction to Client/Server Systems: A practical Guide for Systems Professionals*, John Wiley and Sons, Inc., 1993.

[9] Atre, S., "Train Or Fail: The Hidden Side Of Client/Server", *DBMS* (7:11), October 1994.

[10] Edelstein, H., "Unraveling Client/Server Architecture", *DBMS* (7:12), May 1994.

[11] Deitel, H., Deitel, P., and Nieto, T. *Visual Basic .NET How to Program*, 2nd Edition, Prentice Hall, 2002.