

A Framework for a Web-based Group Decision Support System

A. Nuri Basoglu, Osman Darcan, and Ozgur Ozdemir
Bogaziçi University, Turkey

basoglu@boun.edu.tr, darcan@boun.edu.tr, ozdeozgu@boun.edu.tr

Abstract

Web based systems has started to broaden their span of a applications. Implementing Group Decision Support Systems (GDSS) are now more likely with higher speed and country-wide. In this study, a web based GDSS framework has been designed, a pilot study has been developed and demonstrated on car selection process.

Keywords : Group Decision Support, Web-Applications, Groupware, Java

Introduction

Information technology has transformed the business world, and every workplace. One of the most recent areas that technology has affected the workplace is with the introduction of computer-mediated communication systems in organizations. Recently there has been a rise in use of group decision support systems (GDSS), which are systems that are supposed to help facilitate group interaction. GDSS is commonly referred to as an Electronic Meeting System or groupware and may be defined as a collection of software, hardware, and procedures designed for the automated support of group activities. Deirdre defined it as "electronic facilitator of collaborative synchronous and asynchronous work communication which combine computer technology, video, audio, and telecommunications services" (Deirdre, 1995).

Acquiring relevant data, finding solution approach for multicriteria problems, enabling group discussion and agreement on results, and managing the whole process under an integrated architecture seems to be important issues of today. Software development methodologies and tools

have been under heavy investigation in the last decade. An increasing amount of work has been carried out to build generic and reusable solutions in design of Decision Support Systems and Groupware. Various commercial and research products have been developed (Mockler, 1992). Power indicates multiplier effect of web on power of DSS applications (Power, 98). Byun has shown implementation of Analytic Hierarchy Process (AHP) for automobile selection with inputs from a group of decision makers (Byun, 2001). This study is an augmentation of wDSS (Basoglu et al, 2000) that attempts to propose architecture and establish a web-based software system that enables groups and individuals make decisions in a more structured fashion.

Decision Support Systems

A DSS is a computer-based system that supports choice by assisting the decision maker in organizing information and modeling outcomes. DSS aims to provide help in formulating alternatives, accessing data, developing models, and interpreting their results. One can also select options and analyze the impacts of the selections (Vicky, 1997).

Extending decision systems toward GDSS needs extra effort. Groups systems are part of highly social systems, which envelops computer-based information systems (Figure-1). Groupware practices demonstrate different advantages and disadvantages over conventional approaches that have significant effect on success of the applications. However social nature of groupware is beyond our study. Basic difficulties encountered in developing groupware are summarized below.

Material published as part of this proceedings, either on-line or in print, is copyrighted by the author with permission granted to the publisher of Informing Science for this printing. Permission to make digital or paper copy of part or all of these works for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage AND that copies 1) bear this notice in full and 2) give the full citation on the first page. It is permissible to abstract these works so long as credit is given. To copy in all other cases or to republish or to post on a server or to redistribute to lists requires specific permission from the author.

Our Approach to Group Decision Support Systems Design

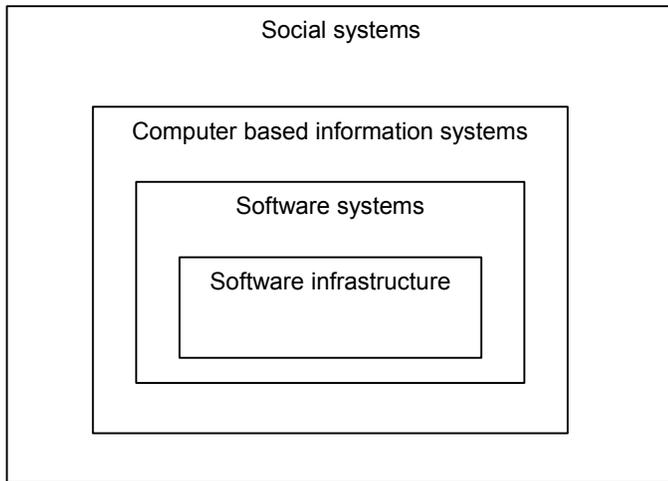


Figure 1: Hierarchy of systems

- Design is complex as groupware users are different in background and roles.
- It is challenging to design an interface that can meet all users' requirements.
- Group dynamics is difficult to understand and draw general conclusions
- Group processes are often uneven and context sensitive and disclose over a relatively longer time frame than the individual activities.
- Group behavior cannot be generalized to other groups.
- Each group is different and a group's behavior is highly influenced by the observation conditions.

It is well accepted that DSS applications need some modifications since the information needs of decision makers may change over time. Hence, whatever tool or method is chosen for the building of a DSS, it must be one that adapts well to changes in the databases accessed, the models used, and even the user interface (Rivett, 1998). On the other hand, object-oriented methods seem to provide the best groundwork for systems that need to be changed over time. The most important characteristic that makes the object-oriented methods valuable is reusability so they provide a platform for faster development and maintenance. (Vicky, 1997)

Apart from the main storage, our approach consists of mainly five components as depicted in Figure 2. These components are **data collector**, **model manager**, **user management**, **criteria specification feeder**, and **display**. In addition to these components, the design incorporates storage elements such as "cache type" database used by data collector and modelbase of the model management. Modeler plays the role of a person who designs and develops the model. The user groups simply represents the different users participating in different problems that interact with the system for a solution space. Group manager module helps the domain leader to arrange group decision activities

The Model Overview

The model manager, which is composed of **data composer** and **model evaluator**, is responsible of organizing data representation and performing deduction process on available data. Specialized task of the data composer is to play the role of a bridge between **data collector** and **model evaluator** (Figure 2). The data composer obtains data from the data collector and realizes conversion, formatting, and representation subtasks. After processing the data, the data composer sends the processed data to the evaluator module. As to the evaluator, it fetches the criteria information of the model, it evaluates these data and it creates the result set. This result set is then sent to the display component in order to show the alternatives that may be appropriate for decision makers. Based on the information available, the end users can make changes on the result set by modifying the criteria vector in order to perform what-if analysis.

Data collector generally extracts raw data from the external sources such as a web site or remote database and accepts inputs from the user. It then transmits these data to the data composer module as the latter requests them. As depicted in Figure 6, our initial design of the data collector component is composed of three main objects such as *source catalogue*, *controller*, and *get_data* objects that work together to extract the required data.

Framework for Web-Based Group DSS

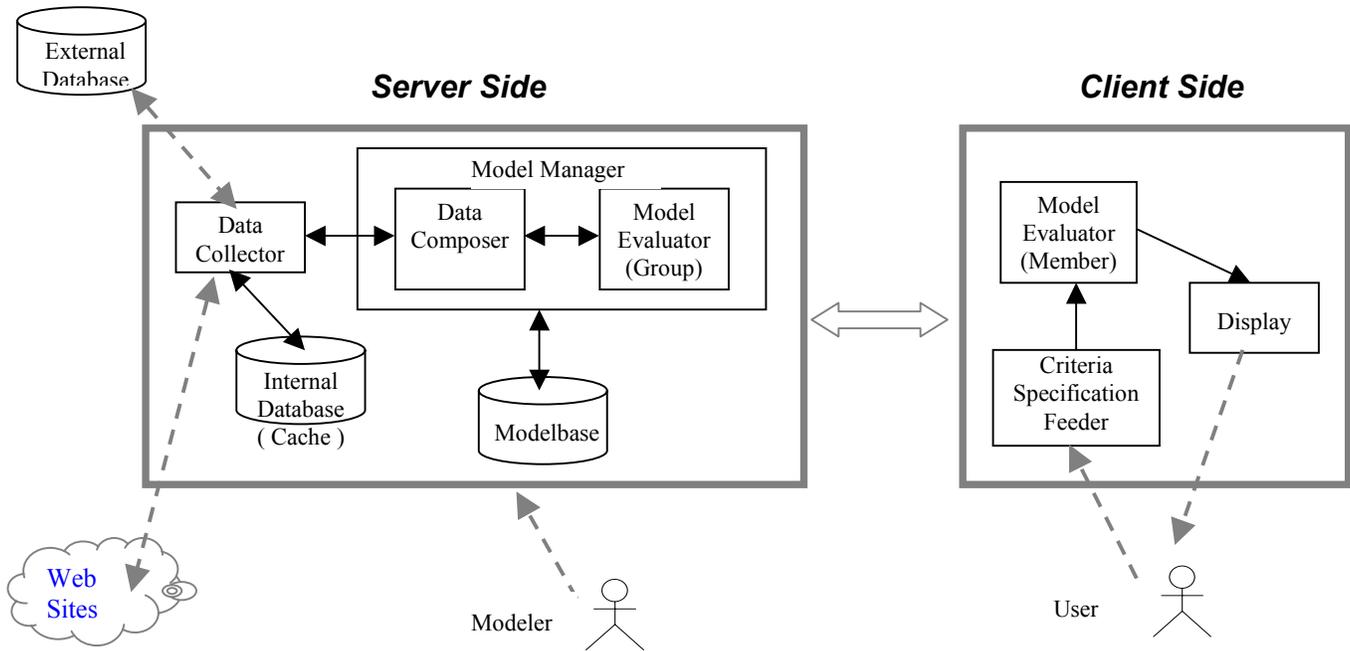


Figure 2: Schematic View of GDSS System

Decision Process Flow

Roles of users and basic steps of decision process may be:

- The group leader determines a domain (car selection, laptop selection etc.), alternatives, group members, and their relative weights.
- Each member picks a subset of model criteria among the whole set, then determines his personal weights (Figure 4 and 5).
- Acquiring appropriate data for each alternative and criterion is compiled by the system, technical staff or by a special team.
- After sufficient data or opinion is gathered, result of 1st tour is produced by running the system and disseminated to the decision team. A sensitivity analysis is performed by altering the weights (Figures 7 and 8).

Same or different members may contribute to alternative generation, criteria and weight determination and selection tasks. Additional tours may be carried until all or majority of decision makers agree on the result.

Steps of Decision

Domain and User Definition

Domains, domain attributes and alternatives are defined by the modeler or the domain leader. A user must define him-

self in order to use the system. While registering, the client applet takes username and user password from the user and sends them to the server application. The server takes them and checks if the username is already assigned to another user in the database. If so, the server sends a warning to the client applet to make the user select another username. If the username is valid, the server registers the user in the database and sends an approval to the client.

Group Definition

When a user defines a group in the system, the client takes a group name, group's domain, group leader's username, and sends them to the server (Figure 3). The server checks the group name assigned by the user for duplicate group name definition in the database. If the group name is valid, the server registers the group in the database and sends an approval to the client to take group users' usernames from the user. For a person to become member of a group, he must be already a registered user.

The screenshot shows a Java applet window titled "Define a Group". It contains four input fields: "Group Name" with the text "bogazici", "Domain" with a dropdown menu showing "CAR", "Leader User Name" with the text "darcan", and "Leader Expertise Level" with the text "100". Below the fields are two buttons: "Done" and "Back to Main". At the bottom of the window, there is a warning message: "Warning: Applet Window".

Figure 3: Group Definition

Mechanism of Internal Engine

After a user logs in the system, the server application checks the user’s group’s domain in the database, takes attributes of that domain and send them to the client (Figure 4). The client shows the attributes to the user, receives user attribute preferences and weights then sends all these information to the server application (Figure 5).

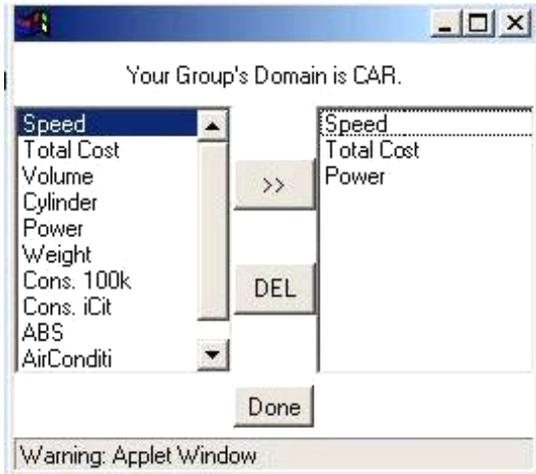


Figure 4: Attribute Selection

All these data from the client makes up a part of model. The server that took the partial model from the client analyses data requirements and looks for needed data. At the first step, it finds alternatives for the selected domain that exist in the database. The server application first looks in a cache database to find values of the alternatives on selected attributes. If the needed data are available and fresh enough then the server application utilizes these values.

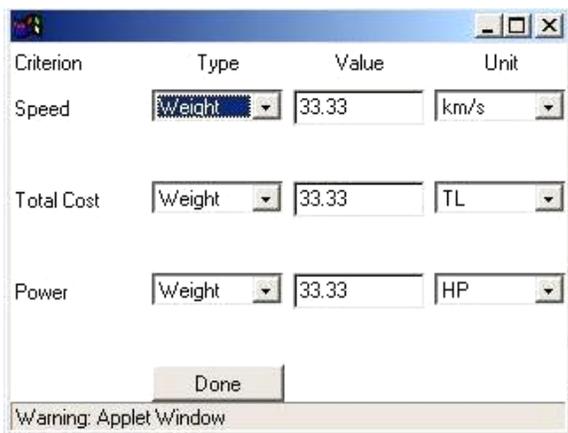


Figure 5: Weight of Attributes

Otherwise the server application attempts to access the external ODBC or web sources. To find the external source’s address, the server application looks in another file in the internal database.

The server application finds the fresh data, reformats and uses it for calculation, also creates a copy of fresh data in the cache database. Once the server application finds all needed data, it calculates relative performances of the alternatives according to the weights that the user assigned and the alternative values on the attributes. Then the server application sends the result to the client to display to the user. The client takes the result information and provides in a graphical user interface to the user. This is the user’s *own model* evaluation not group’s overall evaluation. On the result screen, the user can see his evaluation result and make sensitivity analysis by altering the attribute weights. As the weights are changed, the client normalizes the model weight vector by recalculating all selected attributes’ new weights and runs the model again with new input set. Also the client reflects the new result with GUI on the screen. Once the user makes the final decision on his model, he can save the parameters of the model. The client takes current attributes and weights, and sends them to the server application. As soon as the server application receives data stream and saves in the database for later usage.

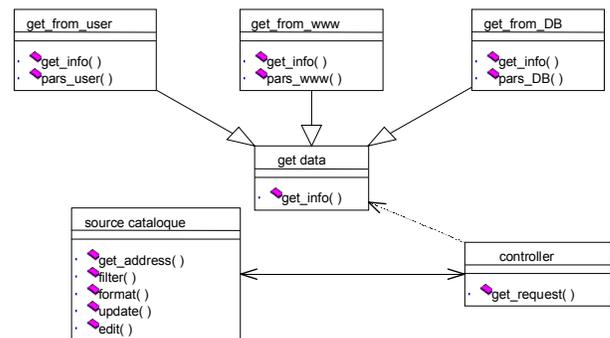


Figure 6: Object View of Data Collector Component

Consolidating Individual Decisions: Group Decision

When a group leader wants to see his group’s overall result, he logs in the system and the client sends the group leader’s request to the server application with the group leader’s group’s name. The server application that took the request looks in the file keeping members’ model evaluation in the database, takes models of members of the

Framework for Web-Based Group DSS

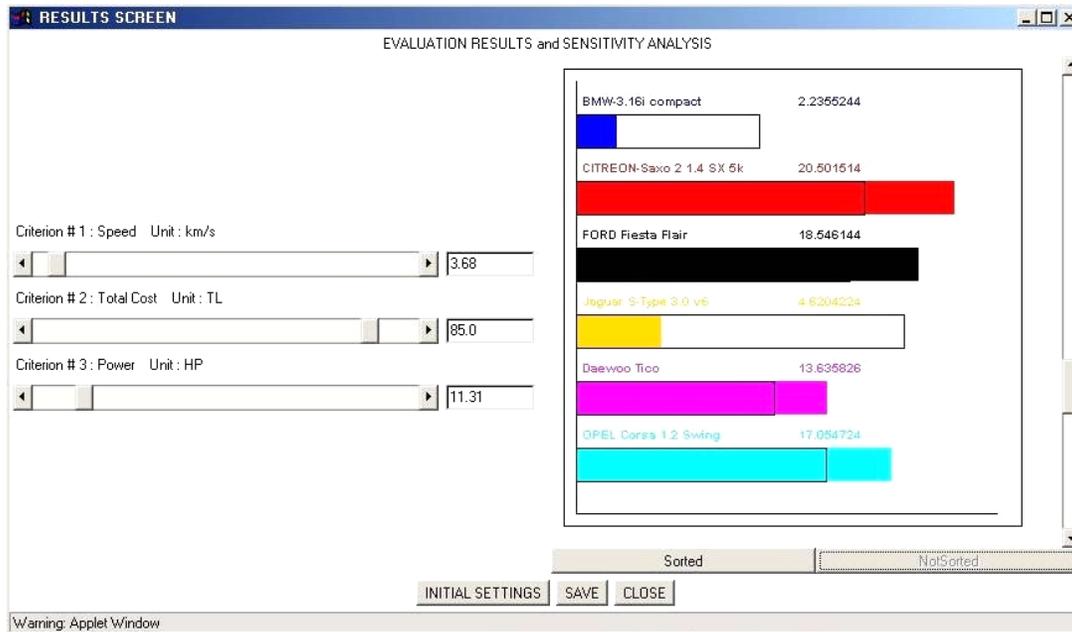


Figure 7: Result of computation based on given attribute weights

specified group, and turns all the models into one group overall model. During this process, the server application takes the members' expertise levels specified in the database into consideration and gives priority to some members' models. Then the server application sends this overall model to the client to show the group leader.

The overall results may be disseminated and later dis-

cussed face-to-face at a round table with the participants. In case the size of group is large, a second tour of evaluation may be initiated where each member may find a chance to review his decision.

A three-tier client/server approach is used in our design. The client is a Java Applet that can be run on any Java-enabled Web browser. In addition to the user interface

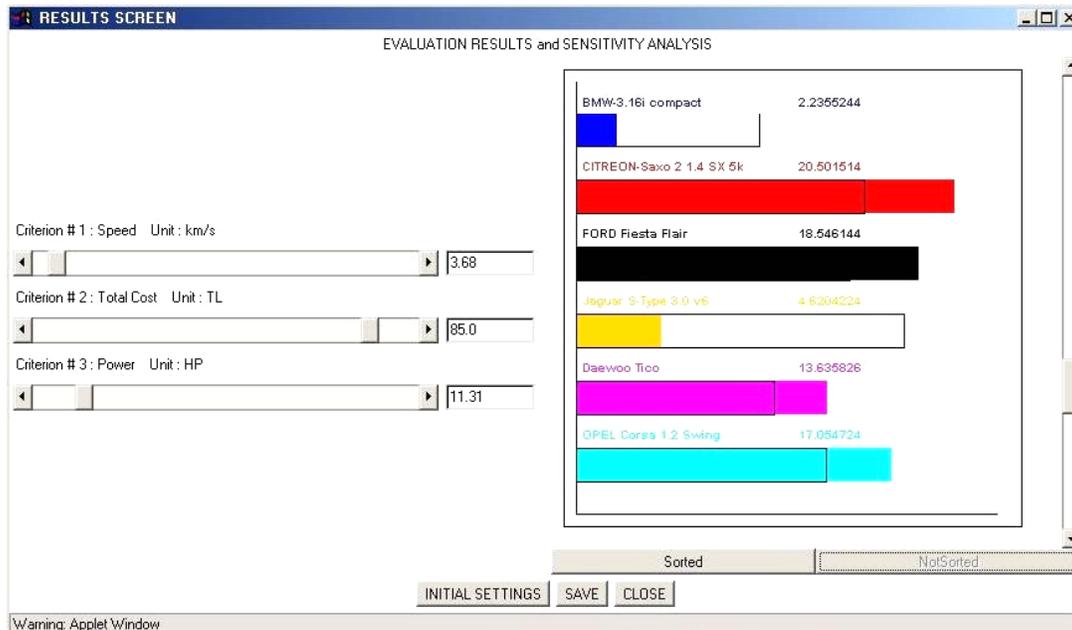


Figure 8: Sensitivity analysis by altering the weights and new Technical Architecture

functions, the client is responsible for knowing about the server, and talks to the server. The server, on the other hand, has the basic functions of accepting calls from the client, returning some results (mostly the data read from the database). The server program is associated with an activation daemon-like process that is responsible for activating the server program as the applet is loaded on the browser. The underlying mechanism that connects two parts is the datagram protocol. As to the database management, in preliminary version of the project the data is stored in a relational database, bridged by ODBC-JDBC connections and data manipulation has been handled using SQL statements

The server application and the client applet communicate via socket connection. They write their message into a character string and put a flag character to first byte of the string. They use a special communication protocol to understand each other. By the help of the flag character, the recipient parses the string and act accordingly.

In the development of this environment, we used the following languages and tools; Java, HTML language, SQL, Visual J++, Web server and database systems.

Conclusion

GDSS could be very useful in the workplace if implemented correctly, with the right control measures. The programs must be used in the right places and with the right kinds of groups to be effective. It makes some groups much more efficient while other groups would only become discouraged and frustrated by it. In the correct setting, GDSS could help a lot of businesses and organizations with problem solving or idea generating.

At present, we have being implementing simple multi-criteria discrete alternative evaluation model. Furthermore, the model has being designed such that different evaluation models can be incorporated to the model frame. With this study, we are trying to fulfill the general properties of a GDSS system such as diversity and flexibility. Data capturing feature is the most critical factor that plays a role in the

success of DSS systems, which we plan to enhance its capacity in further researches.

References

- Basoglu, N. Darcan, O., Baytar, U., Pamir, A.G (2000). "An Object Oriented Decision Support System Tool", *SCI'2000-4th World Multiconference on Systemics, Cybernetics and Informatics*, Orlando, USA, July, 2000
- Byun, D.H. (2001). The AHP approach for selecting an automobile purchase model, *Information Management* 38 (2001) 289-297.
- Deirdre C. D. (1995). Interpersonal Computing and Technology: An Electronic Journal for The 21st Century, July 1995
- Mockler, R.J. (1992). *Computer Software to Support Strategic Management Decision Making*, MacMillan Company, NY.
- Patrick Rivett (1998). *The Craft of Decision Modeling*, John Wiley & Sons.
- Power, D. J. (1998). "Web-based Decision Support Systems", Part I and II. DS*, *The On-Line Executive Journal for Data-Intensive Decision Support*, August 18 and 25, 1998: Vol. 2, Nos. 33 and 34.
- Sauter, V.L. (1997). *Decision Support Systems: An Applied Managerial Approach*, John Wiley & Sons Inc.

Biographies

Nuri Basoglu is a faculty member of Department of Management Information Systems, Bogazici University, where he teaches Database systems, Decision Support Systems and Enterprise Information Systems. Dr.Basoglu has intensive research on knowledge management and group decision support systems.

Osman Darcan is a faculty member of Bogazici University, where he teaches programming languages. His basic interests are Programming Techniques, Object Oriented Systems, and Simulation.

Ozgur Ozdemir is a graduate of Department of Management Information Systems and he is currently employed in SLAT IT-Consulting GmbH.