END-USERS INTERFACE FOR PRACTICAL DECISION SUPPORT SYSTEMS

Do Trung Tuan

Department of Mathematics, Mechanics and Informatics
College of Science, VNU

Abstract. Model of Decision support system is clear. Some systems have been implemented for certain purposes. Although achieved remark results, in study domain problem of organization is influenced from results of others science branches such as Data base Management System, Artificial Intelligent, Information Systems... Then appropriate kernel architecture is necessary for practical applications in which the interface component plays an interactive adaptive role. The paper aims at the End-users Interface in such kernel architecture for constructing other support systems.

Keywords. Interface, Decision support system, Information System, Data base Management System.

I. Introduction

Information Technology (Computer Science) has big intersection with other natural and social sciences. They find a lot of systems that manipulate data and realize mathematical economic models for economic activities. One such system is Decision Support System (DSS) which is represented under different kinds. In recent years, a group decision support system is appropriate approach in distributed environment, in working application for many users [2,4,5,7].

The earlier generations of DSS are Transaction Processing System, Executive Information System. Another definition of DSS is Management Support Systems. The DSS and Expert System could build an effective system in which the machine proposes a solution and alternatives for decision making of managers [7].

Certain application domains of DSS are as follows (i) management science; (ii) Information system; and (iii) interactive system, end users interface. In application systems, the problem is unstructured and semi-structured, i.e. at least one phase among all phases of the problem is not programmed. In fact it is not possible to have enough information, complete information, exact information. Then using a support system is actual interesting and *end-users interface* plays an important role in such systems.

II. Principal Modules of General architecture

From the end users to the meta-base of their computers exist some modules (i) Interface; (ii) Library of models; (iii) Management system that manage data and models; and (iv) Meta-base.

II.1. Interface man-machine

For our opinion, the interactive end users interface is the most important because of its effectiveness on whole system. In this year, a multimedia interface is suitable. A multimodal approach will be developed at next generation of software. In parallel, an oral interface is better, but it requires special equipments of sound processing. Although there are many kinds of 4th generation language, the interface command-response is appreciate in actual systems.

II.2. Library of models

Models such as economic models, mathematical economic models are stored in the base for being called when needed. Generally they are routines which correspond to models of data processing. The system manage them by their identifier and parameters. Information exchanged may be presented under messages between system modules. With such design, users can add/ delete/ update total models or a part of them.

In the DSS models play a role of an alternative method of data processing that permit the interface to present an other available solution. Embedding the library in the Management System is hard for designers. Problems concerning the models in DSS were found in [1,4,7].

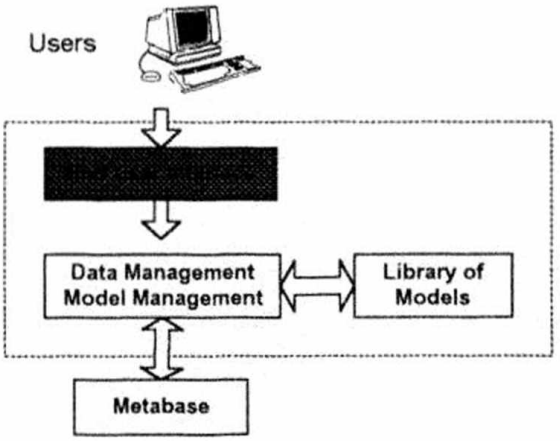

Figure 1. A general architecture of DSS

II.3. Meta-base

Normally a data base is build for at least quantitative manipulation. By the data base, the support systems can realize qualitative, but more difficult and no transparent. However using rules and facts are ready on recent system. A knowledge base is necessary for accumulating and manipulating expert's advise. On a data base management system it is able to extent data to knowledge. A speudo-Relational Model could be proposed actually and further an Object-oriented Model is more appreciated.

II.4. Management system

Beside the function of data manipulation (input/ retrieve/ update/ print), the Management System must be developed to motor engine. Whereas the system is not an Expert System, a mechanism of rule/ fact manipulation is demanded. Control on models is responsible to message manipulation.

II.5. Proposal of a kernel architecture

After some remarks about DSS and its components, after certain results of Turban from [7], of Sprague et al. from [2,3], we proposed a kernel architecture for a practical DSS. In order to implement such architecture, the modules are written separately. They exchange information by messages. Programming language is either C language or Assembly for professional programmers. Furthermore certain tools permit the users to customize the interface and split the big problem into smaller ones.

Concerning the end user interface, the kernel architecture composed of either *the end users interface and the management system* or the *only management system*. The later approach attempts a complete interface with different kinds of command languages. In this paragraph, the Kernel Architecture indicates the Management System. It is a accelerated Database Management System based on Relational Model.

Functions of data manipulation are discussed. It is not new idea :

1. Entry data by data file, forms... from screen or secondary memory. Interactive mode is used. The users present their question under form of SQL command.
2. Retrieve by criteria on attributes of entities. AND qualification norm is used for questions.
3. Update data : inserting, deleting record and modifying data item. These actions are popular in tables of Relational Model.
4. Change data structure : inserting / deleting attributes of relational table and extent user's view.

Other problem having to solve is one hand knowledge representation and other hand model management. The later is satisfied with assistance of message management. A message header, name of routine, parameters are elements of the message.

Figure 2. Message for model management

Knowledge noted here is rule. Because of the capacity of management system in rule/ fact manipulating, rules are simple and related to certain domains. A relational table for rules has two columns (attributes) : IF part and THEN part. The end users consider these rules as records (n-tuples) of relational table. It depends on the treatment way to fire the rules. The IF clause is considered criteria in which system status can be satisfied. A simple form is as qualifier in user's question. The THEN clause is various : determining variable value, exiting to call routine, changing system status... A appropriate strategy to control rules is necessary since the system needs an enough strategy rather than complete one. A simple using of the back chaining, forward chaining techniques in expert system is better.

III. End-users interface

For a multi-purposes practical DSS, we proposed the architecture presented in previous paragraph. An end user interface is worth most interested by following reasons :

- Allowing non-informaticians to access to the support systems;
- Introducing advanced technologies, especially multimedia and multimodal tools;
- Independencies in DSS, in despite of the interface module is in/ out the management system of DSS.

III.1. Interface management systems

An indepedent module with the function as "Interface Management System" is useful in applications with dynamic data where interface builders fail, are really usable for end-users. This remark is same the conclusion in the DSS Interfaces Meeting organized at June 13, 2001 [3]. Such module composed some components for

- Functional processes, within the DSS capabilities/ constraints;
- Transactional interface formats, with emphasis on balance-affecting transactions; and
- Data to be exchanged, in order to ensure specific data element values were meaningful to both DSS and the components asset management system.

Furthermore the interface module is responsible for methods, tools, and techniques for developing the overt user interface of a DSS; managing linguistic, presentation, and user knowledge in a DSS; DSS help facilities; coordinating a DSS's interface events with its functionality events.

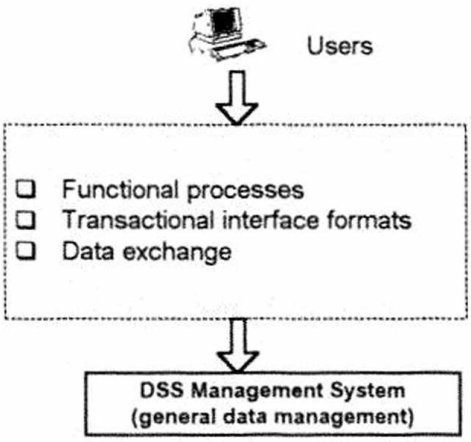

Figure 3. Interface management system in DSS architecture

III.2. Some aspects of DSS's interface

For developing the interface management system, we proposed some aspects of the system that will be respected. Certain aspects were found from Sauter in [2,6].

1. Portable: interface code should be portable, i.e. there should be a way to compile the same code for different host systems such that different look and feels of different platforms can be supported;

2. Matching the needs of users: for commercial application, it is most important to exactly match the look and feel and to exploit the most recent features of the host system. Another dimension to be considered is the way, interfaces are specified: with (i) direct-manipulative interface builders or with (ii) languages that support high-level programming abstractions. Sauter [6] referenced to the research area of user interface management systems in a lot of different systems and approaches, such as of Foley et al. or of Myers.

3. Customize facilities: the facilities are named "Interactive Interface Builders". It seems that the facilities were the best way to construct the DSS interface by interactive graphical manner or by graphical prototypes. Detailed in constructing a DSS interface, it must focus on:

- *Libraries* for standardized dialog objects like buttons, list-gadgets, text-gadgets, gauges that can be interactively composed inside a dialog window. Simple form-based interfaces can be built with minimum effort. However, when more flexibility is needed, there are some problems with this approach.
- *Graphical objects* allows the system designer to build (i) special "actions" are predefined for graphical objects; (ii) actions are realized as generic functions using of host systems; and (iii) additional specific actions can be declared and application-specific methods can be written for predefined actions. The *layout* of the items inside a dialog window must be adapted

when the window is resized. User interface management systems manage pane layout by inserting special panes (layout manager panes) into the pane hierarchy.

- Interface development involves more than interface builders currently support. *Programming* is still required for serious applications.

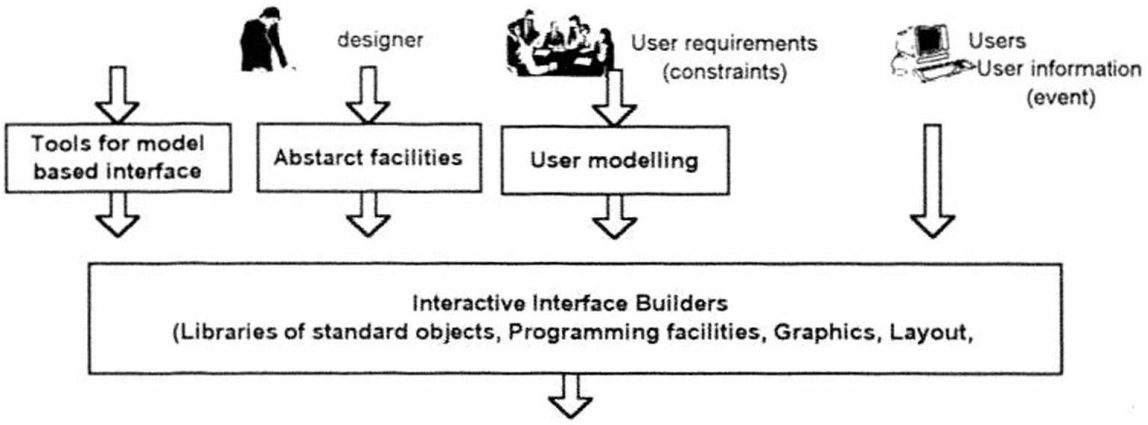

Figure 4. Some aspects concerning to constructing a DSS interface

- *Tools for Model-based Interface* : Model-based interface tools supply explicit models of how an interface should look and behave. Certain aspect concerning (i) Application semantics: objects and operations of the domain of discourse; (ii) Presentation templates: visual appearance of an interface defined by widgets (line, icon, text, menu, button, column, row, table, graph); (iii) Behavior: input gestures to be applied to presented objects for specialized interaction styles; (iv) Dialog sequencing: ordering constraints for commands; and (v) Action side-effects: actions executed automatically after a command (e.g. making a new object the "current" object). Designers prepare knowledge for the functions such as (i) to model attributes of graphical objects; (ii) possible actions defined on them. One goal of the model is to use explicit models to map low-level user gestures onto high-level semantics embodied in the design models and to generate automated (and animated) help facilities.
- *Using abstract facilities*: Abstractions used to model applications and interfaces being introduced by various user interface management systems are quite similar. However, they differ in their concrete environment.

III.3. Intelligent User Interfaces

The area of intelligent user interfaces covers a variety of topics concerned with the application of Artificial Intelligence and knowledge-based techniques to issues of Human-computer Interaction.

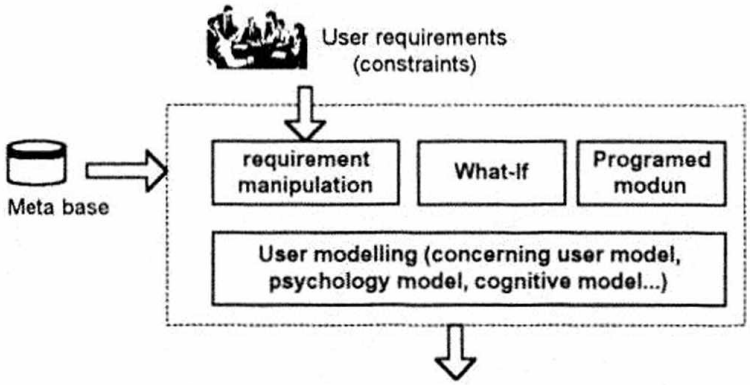

Figure 5. Solution for Intelligent user interface

The design of effective and efficient Human-computer Interfaces becomes ever more critical to overall system performance. Advanced applications are characterized by large amounts of information to be conveyed and understood, complex task structures, real-time performance characteristics, and incorporation of autonomous or semiautonomous agents. The requirements imposed by Human-computer Interaction with such systems exceed the capabilities of conventional interfaces which often fail to reflect the semantics of its users' tasks and problem domain properly. Intelligent user interfaces aim to cope with these serious semantic problems and help users to access information or solve complex tasks by being sensitive to a user's knowledge, misconceptions, goals, and plans.

A beginning version of intelligent interface is programmed module, but later versions can adapt the users need, then propose an appropriate solution based on knowledge in the meta base and user information.

III.4. User modeling

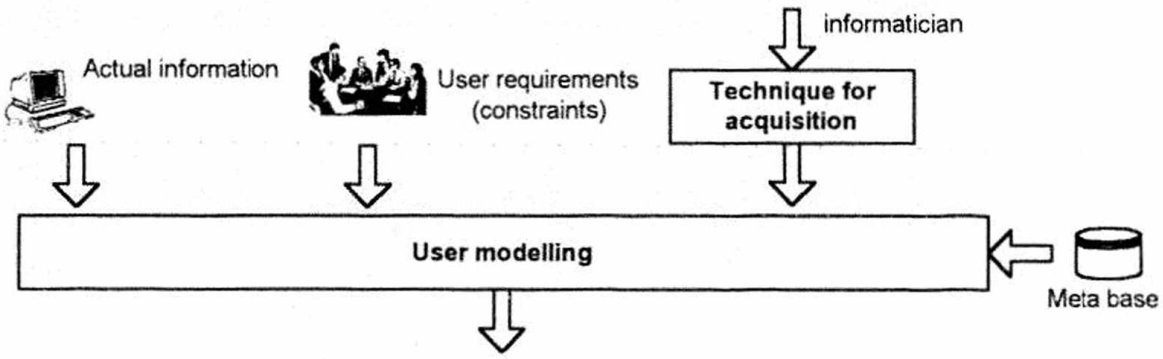

Figure 6. Acquisition of user requirement for user modeling

For user oriented approach in designing DSS's interface, an user modeling is a solution. The internal and communicative behaviour of an interactive computer system is affected by its knowledge about the user to DSS. The knowledge may be implicit in the design of the system, or explicitly available. User-adapted

interaction requires the use of an explicit model of the user, i.e., the knowledge about the user must be explicitly represented and modifiable, and the system has to contain mechanisms to exploit this explicit information to adapt its behaviour to specific users dynamically. A user model is a knowledge source which contains explicit assumptions on all aspects of the user that may be relevant for the behaviour of the system.

III.5. Proposal of End-users interface

For our experiments, following procedure is acceptable :

Procedure for constructing an End-users interface in the practical DSS

The procedure implicates the existence of Data Management System in which da ta and knowledge is manipulaed.

Step 1. Analysing the users requirements

- Preparing tools for surveys to users;
- Acquisition of users' needs;
- Analysing the requirements.

Step 2. Designing End-users interface

- Using "Interactive Interface Builders", normally graphical and interactive tools;
- Proposal of End-users interface;
- Evaluation.

Step 3. Transfering the Interface Specification to Data Management System

- Co-ordinating the interface specification to the Data Management System;
- Evaluation and interface tunning.

Step 4. Access to the Data Management System (requests from End-users interface to the Kernel system)

Step 5. Final evaluation (For accepting the solution of End-users interface).

IV. Conclusion remark

For semi-structure/ non-structure application, DSS is prefered. In the first international conference on DSS organized 1984 in Paris, end user interface was determined as an important component in DSS architecture.

In practical multi purposes in order to introducing support systems rather than general information systems, the end user interface plays the dynamic role and is worth to become an independent one. The interface management system

proposed in this paper is just principal frame, must be detailed for concrete application.

In fact a beautiful architecture is good for presenting, but small simple implementation is acceptable, especially for lower level of industrialization. A DSS can be based on spread sheet as LOTUS 123 and meets user's need. Although in order to develop a bigger system, a kernel proposition is interesting.

REFERENCES

1. Bianchi-Berthouze N., An interactive environment for kansei data mining, 2nd *intenational workshop on multimedia data mining*, August 26th 2001, Sanfrancisco, CA, USA, p 58-67.
2. Brobst S., Rarey J., Five stages of data warehouse decision support evolution, *Teranda magazine*, Spring, NCR corporation, 2001.
3. Reid, C., Decision support systems, *http://www.elsevier.com/inca/publication*, 2000.
4. Sprague, R.H., *Decision Support System, putting theory into practice*, Ed. Prentice Hall, 1989.
5. Sprague H., Carlson E. D., *Building Effective Decision Support Systems. Englewood Cliffs*, N.J.,Prentice-Hall, Inc.: 1982.
6. Sauter, V. L., *Decision support systems*, Chapter 5 : users interface components DSS links, *http://umsl.edu*, 1999.
7. Turban, E., *Decision support systems and Expert systems*, 4th ed., Ed. PrenticeHall, 1996.