A GOAL MODEL FOR THE EVALUATION OF VIDEOTEX SYSTEMS

E. Domínguez, I. Escario, M. A. Zapata, M. J. Lapeña, A. Miguel

Dept. de Ingeniería Eléctrica e Informática
Facultad de Ciencias. Universidad de Zaragoza
50009 Zaragoza (Spain). E-mail: ccia@cc.unizar.es

Abstract. A model for the evaluation of the user goals that are observed in the user interface of a Videotex System is presented here. This concept-oriented model allows specification of the goals that an application shows, or may show, as well as the system behaviour in response to the user action.

Keywords: User interface, goals, evaluation, Videotex, Ibertex, multimedia databases.

1.- Introduction. A Videotex System is an interactive system which may offer a wide variety of services: access to information, transactions (shopping, booking, etc.), amusement games and so on. In designing this type of systems, the user's preferences should have priority over any system-oriented considerations. That is why the user interface is crucial in this kind of systems.

In recent years numerous contributions (see [6]) have been made to the integration between Software Engineering and Human-Computer Interaction, among which the most notable attempt has been the User Software Engineering Method by Wasserman (see [9] and [10]), but a consistent model oriented to specification, design and evaluation of user interface is still missing. The greatest difficulty is that in the user interface several aspects of a very different nature (control, navigation, information structuring, ergonomy, dialogues, user tasks, etc.) have to considered.

From our point of view a suitable solution might consist of developing a theory that provides the necessary tools to construct concreta models which allow specification of every general aspect of user interface, as well as a suitable methodology for design and evaluation. That is why we are developing a theory that intends to abide by the former conditions. This theory, named Theory of Conceptual Modelling [2], is based on the same basic principles as those used by Kangassalo (see [7]) for the conceptual database design.

The purpose of this paper is to present a model, named Goal Model, which has been adapted to the evaluation of the goals that a Videotex application shows, or may show, by means of the user interface. This model is restricted to the Videotex databases that offer

transactional or information services. An example of a hypothetical Videotex application, which we will call Eli_Bookshop, will be used during the exposition.

Within the Theory of Conceptual Modelling, the following stages must be overcome in order to define a concrete model: 1) Defining the perspective of the model, by means of which the general aspects to consider and the way in which they are observed by the evaluator are established; 2) Establishing the properties of the former aspects that are desired to be analysed; 3) Defining which model allows specification of the aspects that have been observed in an application and to evaluate the properties.

The evaluation of a specific application goes by the following stages: 1) Defining the tools to aid evaluation which, usually, will be schemes of data models; 2) Specifying, within the defined model, the aspects that have been observed; 3) Evaluating the properties and establishing the final conclusions.

In this paper, we are going to limit ourselves to presenting the stages of defining a model.

2.- Our perspective of the model. A Videotex database is oriented to displaying information that is stored in it, or to carrying out a transactional service, decided on by the user. The information it displays to the user (e.g. set of data on a book that Eli_Bookshop has in stock) is a structured set of data with non-null information content, that is to say, information that may provide the user with new knowledge. A transactional service (e.g. the purchase of a book) carries an update of the database due to request by the user who accedes to this service. The name we will give to each piece of information shown or services offered by the application will be *final goals*.

The number of final goals that generally offer a Videotex application is high. This is why the designer is obliged to carry out a detailed analysis of the final goals, grouping them into types and generalizing these groups with a view to providing the user with paths of access which will efficiently lead him to the required final goal. To do this, the application displays various identifiers of different types of information or transaction on the screen from among which the user must choose one. *Options* will be the name we give to each type of information or transaction that the application displays to the user as an aid in access to the final goals and in such a way that its selection causes a change in the state of the application which the user can notice.

Menu is understood as being a set of options displayed on a screen. This set represents a certain level of the classification of the final goals. However, due to the restrictions in the number of lines that Videotex video devices have, on some occasions the number of options that might be needed cannot be displayed. Under these circumstances it is necessary to display the set of options on a series of consecutive screens. It therefore seems convenient to define *menu* as a set of options that the application displays to the user, from among which he may choose one, independently of the number of screens that are needed for their display.

To facilitate the location of the option which matches or may match a user's goal, the application displays the options on the screen with certain attributes (colours, spatial distributions, graphic separators, etc.) which differentiate each other and possibly distributes them into various groups. In this way the user can reduce searching space if he first locates the group where he foresees that the option that matches or approaches its goal is there and, later, carries out the search within the chosen group. These groups, extracted through the differences marked by the display attributes can be abstracted like a tree on which the father node represents the screen. Should the menu be displayed on various screens, the trees extracted from each one can be extended thus forming only one tree, by associating a node to the menu and stating as its sons the nodes that represent each screen on which the options are distributed.

Decision tree is the name we will give a tree built up as we explained in the previous paragraph. In this tree the root always represents a menu and the leaves represent options. Every remaining node of the tree represents a subset of menu options which we will call *submenu*. Next we shall consider that an *application goal* is any menu, submenu or option of an application.

3.- Properties and informal criteria for a good design. For the presentation of the model, we are going to indicate only the properties and criteria that, to our judgement, are the most important.

Firstly, it is necessary for the application goals and the user's goals to match; that is, the goals displayed by an application should match those expected by a user. With this idea in mind, two basic criteria are established which must be fulfiled by an application: *Completeness* - which determines that the user's final goals may be reached with the application, and *minimality* - which establishes that the application must not display unnecessary goals (i.e. goals that the user will never expect to reach).

Secondly, the structuring of the goals as well as the syntaxis and semantics of the identifiers must be correct. Related to this type of properties the following criteria can be established: *Friendliness* - which determines the knowledge of the potential users, and *structural correction* - which establishes that the goals the user reaches on choosing a concrete goal must be those expected.

4.- Goal Model. Depending on the properties and criteria previously established, the evaluator must build a model to specify the aspects he observes in a concrete application and, afterwards, evaluate the degree of fulfilment of the rules or criteria. In this section, we are going to present a Goal Model which suits the aspects and rules that, as an example, were previously described. The basic elements of the model are concepts, where a concept is defined in terms of the Universe of Discourse and is identified by a name. A concept is used in order to represent the abstraction of any observable fact or event.

For example: 1) MENU is the abstraction of a complete set of consecutive screens in

such a way that every one of them contains options and the user can choose one, and only one, from among all the displayed options; 2) ADVANCE is the abstraction of one action or a sequence of actions which take the user from one goal to another without the application presenting a third goal between them. Notice that both concepts provide formalized notions about certain aspects of user interface, which are independent of a concrete application. Nevertheless the former abstracts a possible entity noticeable in user interface while the latter represents a certain beaviour of the user/application system. The concepts of the former type are named *statics* and those of the latter are called *dynamics*.

A mechanism is needed to specify the aspects that are observed in a concrete application. For example, in the application Eli_Bookshop the existence of a screen named "General Menu" is observed, on which the application displays a set of options from among which the user can choose one. This screen can be interpreted as a new concept, abstracted by MENU, which we will call Main_Menu. The meaning (or definition) of Main_Menu could either be the above-mentioned or it could be extended with the enumeration of the options that it displays The Theory of Conceptual Modelling provides a builder named *concept value* which is applied to those concepts that are independent of a concrete application, and which associates to them a concept representing a fact or event, observed in a concrete application, which is abstracted by the given concept. Thus, Main_Menu is a value of the concept MENU.

Sometimes, it is necessary to describe types of specific properties of the static and dynamic concepts with the aim of evaluating more accurately what is observed. To do this, new concepts named attributes are defined and are associated to the concept which is described by them. For example, the attribute APPLICATION_NAME, which is the abstraction of the identifiers with which the application presents a goal, can be associated to the concept MENU.

The attributes are also concepts independent of a concrete application, so the builder concept value can be applied to them. For instance, a value of APPLICATION_NAME is "General Menu" which is the name that the application associates to the value of MENU which we have called Main_Menu.

When any aspect of user interface is to be evaluated, it is necessary to accurately define each and every type of aspect that is to be considered. The specification of these types is performed through the named Support. A *Support* is a set of concepts, independent of the concrete applications, by means of which each and every type of aspects to be considered during the evaluation is represented.

The static and dynamic concepts of the Support are structured as trees in such a way that a concept A is a son of a concept B if, and only if, A is a specialization of B. In this sense, a Support is a series of hierarchical trees of concepts in which leaves make up a set of concepts to be used in specifying the observable, whereas the rest of the concepts represent generalizations of the previous, providing a classification which will help the normalized analysis of user interface. In order to define static and dynamic concepts of a concrete Support,

the Theory of Conceptual Modelling provides the specialization and the generalization of concepts as abstraction mechanisms.

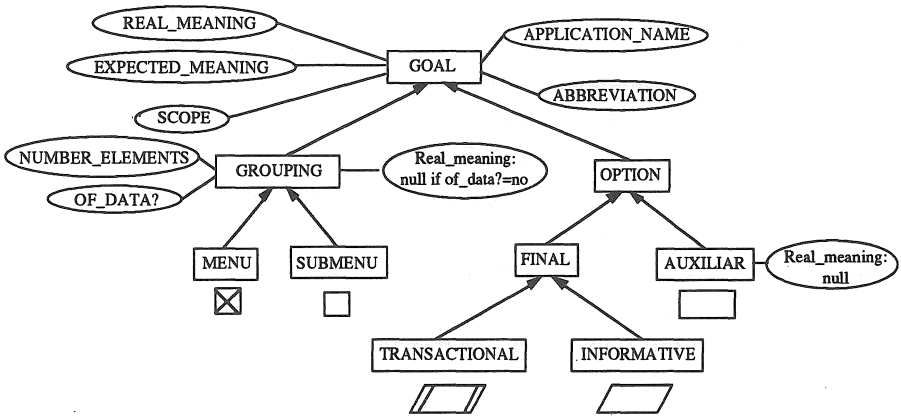

Figure 1.- Goal tree

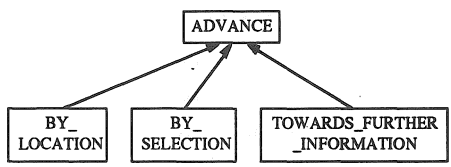

Figure 2.- Advance tree

The Support can also contain attributes. The association of the attribute to the concepts that they describe goes by the following principles: 1) *Inheritance* - it is considered that an attribute associated to a concept is associated to all descendants of this concept, too; 2) *Null value* - attributes able to take a null value are allowed; 3) *Constraint* - when a constraint is imposed on an attribute, this is spread to all descendents of the concept on which this restriction is specified. Other basic principles that Support has to comply with too, are: 1) *Completeness* - every fact or event that the evaluator wants to observe must be abstracted by some concept of the Support; 2) *Classification* - every fact or event that the evaluator wants to observe must be abstracted by one and only one of the terminal concepts of the Support.

The Support associated to the Goal Model consists of a tree of static concepts, named *goal tree* and a tree of dynamic concepts, named advance tree. The goal tree (figure 1) classifies the different types of goals considered in user interface and the advance tree (figure 2) represents the different types of actions. In figures 3 and 4 the definitions of the concepts of the Goal Support are shown.

NAME	MEANING
OPTION	is the abstraction of an entity that the application displays using a name whose meaning is either a possible type of information or modification on the database, or a criterion for access to one of the previous types, and such that its selection produces a change, noticed by the user, in the state of the application.
FINAL OPTION	is an option whose selection by the user produces an access to database which is noticed by the user before reaching the following goal
TRANSACTIONAL	is a final option which carries an updating of the database
INFORMATIVE	is a final option which does not carry an updating of the database
AUXILIARY OPTION	is an option which is not final
GROUPING	is the abstraction of a set af actions represented as a non-terminal node in a particular decision tree
MENU	is a grouping represented as the root in a particular decision tree
SUBMENU	is a grouping not represented as the root in a particular decision tree
GOAL	is the generalization of the concepts OPTION and GROUPING
ADVANCE	is the abstraction of one action or a sequence of actions which take the user from one goal to another without the application presenting a third goal between them
BY_ LOCATION	is an advance defined by a mental action of the user in the menu which he performs in order to locate a goal, provided that this action is represented as an edge of the decision tree of the menu
BY_SELECTION	is an advance defined by a control action which consists of the selection of an auxiliar option
TOWARDS_ FURTHER_ INFORMATION	is an advance defined by several actions, where the first one is the selection of a final option, i.e., it produces an access to the database noticed by the user before reaching the following goal

Figure 3.- Static and dynamic concepts of the Support

NAME	MEANING
APPLICATION_ NAME	is the abstraction of the name given to a goal by the application; it admits null value
ABBREVIATION	is the abstraction of the abbreviation associated to a goal by the analyst to identify it in the diagram and in the tools that aid evaluation
SCOPE	abstracts the fact of whether a goal is accessible to any user (public) or not (private)
EXPECTED_ MEANING	abstracts the meaning that, to the judgement of an analyst, a user would associate to a goal in the moment he identifies it on the screen
REAL_ MEANING	abstracts the meaning of the access to the database associated to a goal; it admits null value
NUMBER_ ELEMENTS	abstracts the number of sons that a grouping has on the decision tree
OF_DATA?	abstracts the fact that at least one son of a grouping represents a value of the database (yes) or none of them does (no)

Figura 4.- Attributes of the Support

In figure 1, shown next to the Support are the graphical describers associated to each of its terminal nodes. These describers are used in connection with another essential component of the model, the Diagram, whose purpose is to represent statically the behaviour of the user interface. Given a concrete application, the *Diagram* is made up of all the values of the concepts specified in the Support which are observed in this application. The graphical representation of the Diagram is carried out by means of a type of graph in which the nodes are values of observed goals along with the values of their attributes and an edge between two nodes represents an advance value between the goals representing these nodes. Every node of the graph will be represented by using the describers corresponding to the concept of which this is a value. In figure 5 we present the Diagram associated to the application Eli_Bookshop, in which, for simplicity, not all values of the attributes have been presented. As it can be observed, this diagram does not fulfil some of the criteria which have been established for a good design. If the stages of evaluating an application are followed, this would be detected.

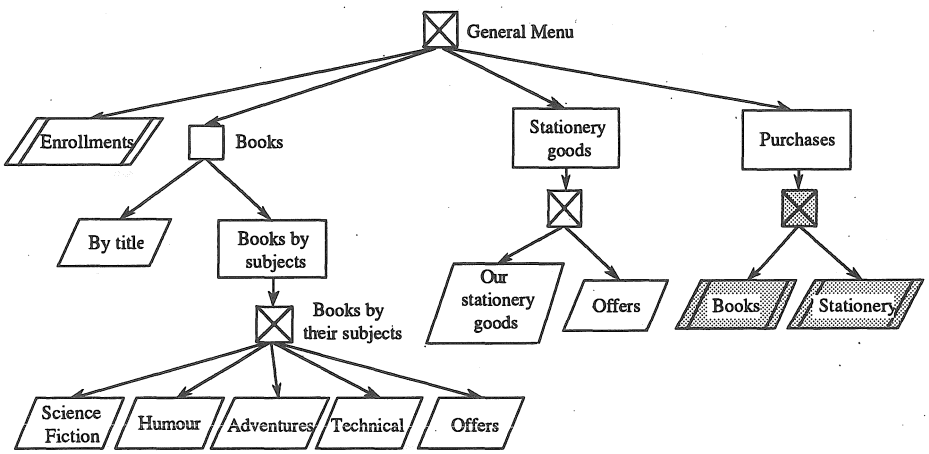

Figure 5.- Diagram of the Eli_Bookshop

5.- Final remarks. The evaluator, depending on his perspective and the particular aspect he wishes to analyse concerning the goals of an application may extend or modify the presented Support. In fact, owing to space limits, this support is a simplification of the one we use in real cases which has been experienced through the evaluation of Ibertex services (Ibertex is the Spanish Videotex Service). Following the Theory of Conceptual Modelling, specific models have also been developed, [3] and [4], for evaluating other aspects of user interface.

Acknowledgements: We wish to thank A. Francés, J. Rubio, A. Cebollada and J. Lloret for their very useful comments during the preparation of this article. We would also like to thank J. L. Rebollo for their invaluable advice on the Ibertex System. This paper has been supported by Telefónica de España S.A. y CICYT (Comisión Interministerial de Ciencia y Tecnología; Project INF89-0879).

References

[1] S.K. Card, T.P. Moran, A. Newell, *The Psycology of Human-Computer Interaction*, Lawrence Erlbaum Assoc., London 1983.

[2] E. Domínguez, I. Escario, M.J. Lapeña, M.A. Zapata, A Theory of Conceptual Modelling, to appear.

[3] E. Domínguez, M A. Zapata, A Model for Human-Computer Interaction, to appear.

[4] E. Domínguez, M. A. Zapata, I. Escario y A. Cebollada, A Control/Navigation Model for the evaluation of Videotex Systems, to appear.

[5] I. J. Elson, Design Trends in Videotex Systems, *Videodisk and Optical Disk* 4(5), 1984, 357-364.

[6] M. Helander (Ed), *Handbook of Human-Computer Interaction*, North Holland, 1990.

[7] H. Kangassalo, Foundations of Conceptual Modelling: A theory construction view, in *Information Modelling and Knowledge Bases*, H. Kangassalo, S. Ohsuga, H. Jaakkola (Ed), IOS Press, 1990, 19-35.

[8] W.A. Simcox, A Framework for the inclusion of Human Factors in the Design of Videotex Systems, *Information Design Journal* 3, 1983, 215-230.

[9] A. I. Wasserman, Developing Interactive Systems with the User Software Engineering Methodology, in *Proceedings Interact-84*, B. Shacked (Ed), vol. 1, 1984, 472-477, North Holland.

[10] A. I. Wasserman, P. A. Pircher, D. T. Shewmake and M. L. Kersten, Developing Interactive Information Systems with the User Software Engineering Methodology, *IEEE Transact. in Soft. Eng.* 12(2), 1986, 326-345.