

## User interface modeling

**RAVINDER G. REDDY**

Department of CSE, BR Engineering College, Hyderabad (India).

(Received: June 03, 2010; Accepted: July 10, 2010)

### ABSTRACT

This paper focuses on the present user interface modeling techniques for capturing the user needs to perform in order to reach his goals. Newly conceived HCI modeling languages need to foresee their role as members of the family of languages that constitute the UML representations for software design, due to their wide acceptance by both researchers and practitioners. This paper extends one such modeling language (MoLIC) and presents an overview & comparison of the trends in user interface modeling in UMLi.

**Key words:** User interface modeling.

### INTRODUCTION

The goal of UI is to minimize the possibility to overlook UI requirements by developing them early on in the design process and by using them as a comparison with all developed components of the system model.

In software development, designers use model that help in object-oriented development, understand, organize and represent the system's architecture and functionally. UML provides us with a large set of models to represent complementary aspects of the system. Use cases, class diagrams, sequence diagrams and activity diagrams are perhaps the most widely used UML models in the initial design stages.

From the user's point of view, the user interface is a single unified artifact. In order to design the user interface, HCI designers adopt decomposition strategies that are different from those adopted in SE, deriving task models and storyboards.

User interfaces design can be separated into four components. In the conceptual design, the

principal application concept like objects, relationship and metaphors are developed. In the functional or semantic design, all operations on all objects of the user interface are defined with their input and output information. Sequencing or syntactic design defines the ordering of inputs and outputs to the system.

GUI layout (also called screen design) is the action of methodically creating the visual presentation of a user interface by spatially arranging elements like lines, boxes or text on a computer screen.

The typical components and the principal development procedure of a model-based interface development environment are shown in Fig. 1. The central component of each MB-IDE is the interface Model which includes different declarative models.

### Model based design patterns

Design is about making choices, concerning among others WHICH patterns of action the user should be able to perform and HOW the design elements are selected and composed to support this behavior.

The development techniques for constructing UI models and generating UIs from these models have been studied since interface management system proposed during 80s. Model based user interface development environments (MB-UIDEs) have provided additional insight on the construction of user interface models.

**UML & UI**

However the use of different notations in MB-UIDEs has led us to the idea of modeling notation has attracted the attention of both academic and put in practical use in the software industry. In the realm of SE, this approach has been working fine for most purposes the summary of UML diagrams used as resource of model UI elements.

Use cases describe the way a system will be used, but they don't specify the system user interface (UI). UML has a rich set of constructors complete enough to model the architectural aspects in user interface but may not be as simple as expected and desired.

The basic UML diagrams do not differentiate between what is internal to the system and what is to the user interface between what is

internal to the system and what is to the user interface. In a class diagram, we cannot distinguish which attributes or methods will have a representation at the user interface. UML does not provide clearly the relationship between use cases and activities. UML does not have a notation to describe of UI interface. UML does not provide a relationship between classes and activities to identify which UI is related to each activity.

**Integration of UI modeling and UML**

UML does not provide designer with modeling languages and associated tools to explore the relevant HCI aspects of interactive software. Modeling languages that express the HCI concern do necessary to bridge both understand and communication gaps between the areas of SE and HCI. From an HCI perspective, one of the major drawbacks of UML is that most of its models concern the system only, leaving most decisions about the user interface to the later stages of development.

Several researchers have investigated several integrating interface modeling techniques with UML.

- a) Assess UML models for use in interface

Model	Content
Task	What the user does or wants to do and why
Domain/data	Concepts, object and operations etc. describing the domain
Dialogue/conversation	The structure of dialogue between human and computer
Concrete interaction	The structure of output and mapping of data and operations
Concrete interaction presentation	Means of inputting data and invoking operations
Control	The application's services that task performance rely on
Platform	Input and output capabilities of the device
Environment	The physical and cultural context of interaction
User	Characteristics and abilities of end user not captured by other models

User Interface Elements	UML Resources	
Domain Model	Class diagrams	b) modeling, comparing them with a collection of specialist interface modelling notations.
Task model	Activity diagram	
Presentation Model	Class interaction and object diagrams	
Exception handlers	Class and activity diagrams	c) Another approach lets comprehensively

model Web applications, but this approach is less conceptual than the other proposal.

Notable trends in adopting UML with added notations to derive user interface (GUI) scenarios for a better design under HCI include UMLi (Silva), MoLIC (Paula) to mention a few.

### **UMLi**

UMLi is probably the most technically mature proposal for interface development in UML. In UMLi, model task using extended activity diagrams rather than by incorporating a completely new notation into UML. UMLi also addressed the relationships between uses cases, takes and views and thoroughly addresses the data on which they act. UMLi also provides specialized visualizations for abstract presentation models. Use-case diagrams in UML. UMLi provides a notation for a set of macros for activity diagrams that we can use to model behavior categories usually observed in user interfaces. UMLi provides an alternative diagram notation for describing abstract interaction objects.

The UMLi metamodel fully integrates with UML. And we can build and integrate UMLi models with other UML models in an extension to the Argo UML toolset.

However, UMLi interface diagrams are essentially UML class diagrams that clarify the purpose of individual abstract components and the containment relationships between different components.

### **MoLIC**

Based on Semiotic Engineering MoLIC represents interactinos as threads of conversation users may have with the system, without yet specifying in detail the storyboards or the user interface itself. MoLIC is an important resource for building the storyboards or rigorously specifying the user interface. Each thread of conversation represents courses of action to accomplish a certain goal or take remedial action when a communicative breakdown occurs.

MoLIC was devised to explicitly represent all possible interactions, allowing desingers to inspect the model for inconsistencies. A MoLIC diagram may be seen as a graphical view of the whole set of scenarios or use cases, from the user's perspective.

MoLIC has both an abbreviated and an extended diagrammatic representation. The extended MoLIC diagram represents not only dialogues, but also what can be talked about in each dialogue. This is represented by the signs at the user interface that either the user or the system manipulates at each moment. Here, we use the term sign to denote any given element at the user interface to which a user may attribute meaning with respect to his/her goal or task, or to the application itself.

### **Extended MoLIC: (Proposed)**

Extended/added modifications include

1. To continue both user and goal as one entity.
2. To identify the submit user utterances by another line format.
3. To include the work elements (UI elements) in the users topic or task.

MoLIC appears to be a natural family member of newly conceived HCI modeling languages since many consistency checks seem to be possible between MoLIC to the UML family of modeling languages will probably require changes in several existing UML diagrams to allow for its proper integration.

## **CONCLUSIONS**

A number of models based UI development systems have been built which have limited modeling capabilities and narrow target applications areas. By systematic comparison of various UI design patterns being developed by object oriented design metrics can contribute to the evaluation of user interface design and in the development of UI software as a reference point for global designs decisions and would be a resource for deriving both HCI and GUI.

**REFERENCES**

1. Maira Greco de Paula D.J. Barbosa, Relating Human-Computer Interaction and Software Engineering Concerns: Towards Extending UML Through an Interaction Modeling Language, PUC0RioInf.MCC 18/03 Julho, (2003).
2. Barbosa, S.D.J., Paula, M.G., Interaction Modeling as a Binding Thread in the Software Development Process. Workshop Bridging the Gaps Between Software Engineering and Human-Computer Interaction, at ICSE 2003, Oregon, USA. May (2003).
3. Paulo Pinheiro da Silva Norman W. Paton, Improving UML Support for User interface Design: A Metric Assessment of UML.
4. G. Booch, J. Rumbaugh and I. Jacobson, The Unified Modeling Language User Guide. Addison-Wesley, Reading M.A.