MOBI-D: A Model-Based Development Environment for User-Centered Design

Angel R. Puerta and David Maulsby
Stanford University
Stanford, CA 94305-5479 USA
+1 415 723 6979
{puerta,maulsby}@smi.stanford.edu

ABSTRACT

MOBI-D (Model-Based Interface Designer) is a software environment the design and development of user interfaces from declarative interface models. End-users informally describe tasks and data, from end-users, from which developers construct formal models of user tasks and domain objects. The system supports development of presentation and dialog specifications from such models, and allows visualization of interface designs as units encompassing all relations and dependencies among the elements of task, data and user-interface specifications. MOBI-D is the first development environment to define an interface model as a comprehensive conceptual object, to identify an interface design as a declarative component of an interface model, and to establish a development cycle based on such a model. The sharable nature of the interface modeling language of MOBI-D, along with the open architecture of its system opens the door for many research areas in HCI to explore the benefits and potential of using interface models.

Keywords

Model-Based Interface Development, User Interface Development Environments, Interface Design, Interface Models, User-Centered Design, Task-Based Design

USER-CENTERED DESIGN

User-centered design is a long-established goal in interface development. Interfaces designed in have shown increased user acceptance and usability. However, software support for user-centered interface design in an integrated environment has lagged behind and is not currently available.

One reason for this problem is that designing user interfaces requires relating very abstract concepts, such as a user task, to very concrete interface elements, such as the widgets in a dialog box. Current tools for interface development do not support that process, and tend to focus on widget layout. The question of what part of a user task is accomplished via a given interface component is answerable only in the designer’s mind, or perhaps via some loosely connected documents.

In this paper, we introduce MOBI-D, a development environment that uses declarative interface models as conceptual design units. These units allow the construction of interfaces by defining and relating user tasks and domain objects to presentation and dialog interface components.

THE MOBI-D ENVIRONMENT

There are two main elements in the MOBI-D software environment:

- *An interface modeling language*. This language defines the organization, components and relationships of interface models.
- *An open software architecture*. The architecture integrates interactive tools for the definition, editing, and refinement of interface models, and for the design and development of interfaces based on those models.

The environment defines a development cycle for interface design based on models.

The Interface Modeling Language

The modeling language in MOBI-D [1] goes beyond the problem of representing the elements of an interface that has been examined in the past. In addition, it defines the structure, components, and organization of interface models, therefore providing a metalevel view of how interface models are constructed. This capability is crucial to build an architecture for model-based interface development. There are two key concepts supported by this language:

- An *interface model* is a set of one or more of the following components: User tasks, domains, presentations, dialogs, user types, and designs.
- An *interface* is an organized collection of interface objects derived from the model components.
- An *interface design* is a set of relationships among interface objects in an interface as defined above. It answers the question of how, for example, a given widget relates to a dialog structure, to a presentation scheme, to a domain element, and to a user task.

The MOBI-D Architecture

Users of MOBI-D have access to a number of interactive tools that support the modeling language [2]. There is no need for these users to write in the modeling language.

The architecture is open and allows any type of tool that can read and write MOBI-D interface models. Each type of model component may be edited and refined by one or more tools. Some of the of tools available in MOBI-D include model component editors (e.g., for building user-task models), design editors (e.g., for assigning a widget to a user task), and decision support tools (e.g., for making layout recommendations based on a user-task model).

The Development Cycle

Figure 1 shows the MOBI-D development cycle. All the processes are fully interactive and may involve user participation. The cycle is iterative in nature. Interface design begins with the elicitation of the user tasks. An interactive MOBI-D tool allows the user to enter a textual description of the task. The tool elicits key terms such as objects and actions and guides the user in editing and refining the terms into a structured user task description. Next, The developer takes this description and builds user-task and domain models with MOBI-D model editing tools. The models are then integrated so that domain objects are related to the user tasks for which they are relevant.

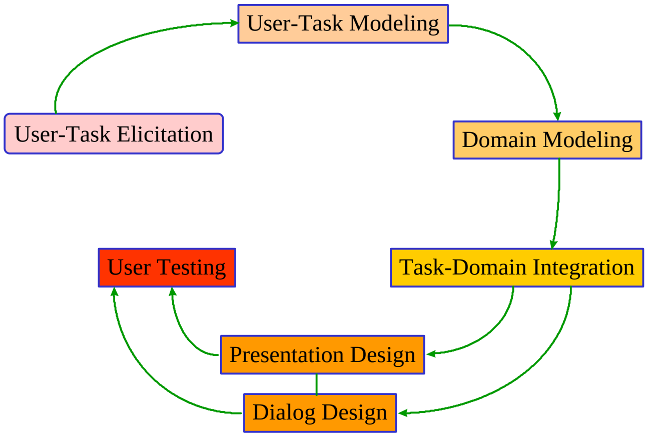

Figure 1. The interactive, user-centered development cycle in MOBI-D.

The decision support mechanisms in MOBI-D use the user-task and domain models to make recommendations for presentation and interaction techniques. These recommendations are displayed to the developer to guide the design and ensure that all task and data elements are embodied in the interface. In effect, Mobi-D walks the developer through the selection and layout of interface components, providing for each subtask a choice of optional components pre-configured for the task data. For example, if the user must enter a number, Mobi-D provides a choice of slider and text-entry widgets with labels and range bounds suggested by the model. The developer could select the slider, position it in the dialog window, resize it and edit its label, color, etc. The resulting interface is tested by the end-user.

Application Areas and Evaluation

We are using MOBI-D to build an Interactive Logistics Map (ILM) interface that permits the visualization and editing of logistics data stored in a distributed geographic information system. Designing such a user interface presents interesting challenges due to the many types of users and user tasks to be supported. MOBI-D is especially helpful here because it centralizes design knowledge in its interface models and facilitates the visualization of the relations between user tasks and interface components.

RELATED WORK

A number of model-based developments systems have been built in the last few years [3]. These systems have limited modeling capabilities and narrow target application areas. MOBI-D is the first of its kind to define interface models as organized and integrated computational units. It is also the first one to offer developers interactive tool support for model-based development instead of a mostly automated approach where flexibility is limited and interface designers are not in control of the design process.

LOOKING TO THE FUTURE

There are many areas that can be potentially impacted in the future by the availability of systems like MOBI-D. We name here the following three:

- *Design reuse.* The availability of complete interface designs in a declarative form allows the reuse, indexing, and processing of designs for new applications.
- *Framework for component-based design.* Objects such as Java applets or ActiveX controls can be made interface primitives and incorporated in interface models. Presentation and dialog design can then take place at a higher level of abstraction than that of current widget toolkits.
- *Runtime functions.* Usability functions and other runtime user-support activities can be linked to a declarative model. This will enable tool-based analysis and decision support for these activities.

ACKNOWLEDGMENTS

This work is supported by the Defense Advanced Research Projects Agency under contract N66001-96-C-8525. Our thanks to Eric Cheng, Kjetil Larsen, Justin Min, and Raymond Chung-Man Tam for their work in this project

REFERENCES

1. Puerta, A.R., The Mecano Project: Comprehensive and Integrated Support for Model-Based Interface Development, in Proc. of the 2nd Int. Workshop on Computer-Aided Design of User Interfaces, Presses Univ. de Namur, Namur, Belgium 1996, pp. 19-36.
2. Puerta, A.R., and Maulsby D., Management of Interface Design Knowledge with MOBI-D, in Proc. of IUI97, International Conference on Intelligent User Interfaces, ACM Press, pp. 249-252.

3. Puerta, A.R., and Szekely P., Model-Based Interface Development, Tutorial Notes, CHI'94, Boston, MA. (available as http://smi.stanford.edu/projects/mecano/model-based.html)