Object View and Interaction Design

Dave Roberts, Dick Berry, Scott Isensee

IBM United Kingdom Limited
PO Box 31, Birmingham Road
Warwick
England

Dave_Roberts@aussmtp.austin.ibm.com

ABSTRACT Several methods are already available for object oriented program design. These methods do not deal with user interface design. The tutorial teaches OVID, a systematic method for designing Object User Interfaces for use by product design teams. OVID is a major step in changing user interface design from art to science. It emphasizes the production of a complete, accurate model that can be used as input to program design methodologies.

KEYWORDS User Interface Design, Object Oriented, Task Analysis

1 INTRODUCTION

In this tutorial you will learn techniques for turning user requirements into design for an object-oriented user interface. The tools and techniques described will help to ensure accuracy and completeness of the model and will generate output that feeds into object-oriented code design methodologies such as OMT or Booch.

- a initial object model,
- views descriptions for the objects,
- new task descriptions using the objects and views,
- interactions within the views
- a more refined object model and so on.

2 TUTORIAL CONTENT

The tutorial begins with a description of the software development life cycle to place OVID in context. It includes a definition of OUI and discussion of the users' and designer's models. These matters will not be discussed in depth. The bulk of the tutorial is devoted to the process of design.

OVID is an iterative method. It takes the output of task analysis and produces:

Figure 1: An outline of the OVID processes

2.1 Task Analysis

The tutorial does not teach task analysis. Instead, it concentrates on how the output of task analysis can be used as input to design and though out the design process.

2.2 Class Diagrams

The core of OVID is class diagrams showing the designer's model. To facilitate transfer of the output of OVID to code development these diagrams are best captured using a tool such as Rational Rose/C++. Examples shown during the tutorial will use Rational Rose/C++. However, exercises will be conducted using pencil and paper.

2.3 Objects

Stage 1 of the process is to find the objects that the user will deal with. Although several users' models may be found the objective is to consolidate to a single designer's model that is appropriate to all the users.

Analysis starts with the most frequent tasks. These are used to find the most concrete objects and the strongest relationships between them. As analysis proceeds the model is expanded to map the relationships for all tasks. In later iterations of the object model the design will express the most significant attributes as objects.

It often emerges that some users think of two classes as distinct when the designer thinks these are variants of the same object. The tutorial gives guidance on this and other modeling dilemmas such as how to represent *many-to-many* relationships.

2.4 Views

The next section deals with how to find the views of the objects that are needed . It also helps the designer decide which views can be treated as standard view types and which require special mechanisms. It goes on to discuss how represent the users as objects in the model can be useful both in the design and code phases of a project.

2.5 New Tasks

The next step in OVID is task synthesis. Tasks must be created that correspond to each of those described during the task analysis. These new tasks document how the users will work with the new system. The objects and views that have already been discovered are used to create the new tasks.

The synthesized tasks are recorded using the same Rose tool and will be used later by the code designers as test cases.

2.6 Detailed View Interactions

The final element within OVID is the use of state diagrams and state tables to provide detailed interaction descriptions for all of the objects and their views. Details are given of how other objects may be discovered during this process. This completes the designers arsenal of procedures used in the design.

2.7 Conclusion

To complete the tutorial a short session discusses: visual design, prototyping, usability evaluation and the transfer to code development.

3 INTENDED AUDIENCE

HCI professionals responsible for turning user requirements into object-oriented interface designs. Background in at least one (but not all) of the following areas is recommended: requirements gathering, user interface design, and object-oriented analysis. This tutorial is about object-oriented user interface design. It is not about object-oriented programming

4 BACKGROUND

The presenters are all members of IBM's User Interface Strategy team. They are currently engaged on development of next-generation user interfaces, style guidance for these interfaces and methods for their deployment.

The tutorial was presented at CHI 97 in March.