INTERACTIVE PROTOTYPING METHODOLOGY FOR A
DOMAIN-DEPENDENT USER INTERFACE

Peter H. Jones
The Analytic Sciences Corp., Fairborn, Ohio

Capt. Douglas A. Popken
AFHRL/LRL, WPAFB, Ohio

TASC is developing the Integrated Model Development Environment (IMDE) under contract for the Air Force Human Resources Laboratory (AFHRL) as a software environment for airbase logistics modeling. This paper describes the methodology for developing the user interface (UI). A UI prototyping process is planned to elicit task-oriented requirements from users of current airbase simulations. An interactive prototyping methodology was developed as an alternative to formal task analysis due to the absence of an existing user base from which to gather task information. This process requires subjects to interact with a prototype UI and to provide feedback to the human factors engineer regarding the usefulness and usability of the UI, and its graphics and concepts.

An initial graphical UI is designed based on analyses of projected user's and expert's conceptual (mental) models of airbase logistics and simulations. A cognitive network method is used for mental model extraction, which is based on knowledge acquisition methods from artificial intelligence. This network of expertiser concepts affords the design of GUI prototypes that map to the expertiser's conceptual model of the domain, thus supporting an intuitive and understandable command and menu structure and graphical representations. Then, interactive rapid prototyping of the GUI using a graphical interface builder is used to capture user concepts early in design, and to provide an opportunity for users/domain experts and developers to work together in the UI design process.

Introduction

The Integrated Model Development Environment (IMDE) is a unique simulation processing system (Clark and Popken, 1991). It will provide a set of tools for logistics analysts to prepare an entire simulation analysis in a single environment, using an object-oriented database management system and the MODSIM II \(^{TM}\) simulation language. IMDE is being designed to interface with four different levels of users: the User level, setting parameters and analyzing predefined problem sets, the Analyst level, to enable selection of model parts from the Premodel database for construction of executable simulation, the Developer level, to support development of new classes within the simulation context, and the Utility level, allowing system management and programming at any level required.

Like many new technology-driven systems without an existing base of users, the IMDE requires a "ground-up" approach to user analysis. In designing user interfaces and user interaction techniques for highly-interactive complex systems, traditional task and requirements analyses do not provide a rich enough source of information about the domain from which to work. Especially in the case of IMDE, which deals with the domain of airbase logistics simulation and modeling, no systems similar to IMDE currently exist from which to infer user requirements, especially in regard to the requirements for the graphical user interface. Currently in this domain, users of simulation tools such as Theatre Simulation of Airbase Resources (TSAR) and Logistics Composite Model (LCOM) employ FORTRAN or SIMSCRIPT programs on mainframe computers, and are not familiar with tools such as IMDE from which any transfer of task information could be made.
The IMDE system is being designed to provide a more productive and maintainable simulation environment to address the known deficiencies of these large batch-oriented software models. The object-oriented design approach used for IMDE will specifically support the graphical user interface design and the capability to build simulations from reusable classes of predefined characteristics.

User Conceptual Model Analysis

An alternative methodology to traditional task analysis was planned for IMDE user analysis. This methodology incorporates knowledge acquisition techniques from artificial intelligence (AI) studies (specifically expert systems), and uses an interactive user interface prototyping process to elicit task requirements from current airbase simulation users. Since IMDE will be an interactive, graphical simulation modeling environment, the representation of a conceptual model for its operation is important. Therefore, the knowledge acquisition techniques are being employed to gather conceptual (or mental) model information relative to this environment.

The mental model acquisition approach is important for such a simulation environment since the purpose of a simulation is to construct, run, and then analyze a simulated model of hypothetical real world situations. The simulation user/analyst's mental models of setting up and running simulations can be considered to directly reflect their understanding (or mental models) of these real world situations. By eliciting and describing the current user/analyst's mental models we expect to gain knowledge of the explicit and implicit tasks and procedures necessary to effectively model airbase logistics scenarios.

Mental or conceptual models are defined as a user's personal, internal representation of a domain of experience or knowledge, related systems, and the systems operations. The extensive mental models and broad range of experience that expert users have to offer in the related domain of non-interactive, non-graphical simulations will provide us with important procedures, concepts, and domain-specific tasks which will support system design. This methodology will provide useful data for design of the user interface and the object database based upon user and expert knowledge of the domain and its simulations.

To solve the problem of gathering such conceptual information from logistics analysts, three knowledge acquisition approaches were considered. Verbal protocol analysis (Ericsson and Simon, 1980) was selected to provide a means of gathering higher-level user-oriented system requirements and the users' problem-solving approaches used in thinking through a logistics simulation. Verbal protocol analysis is essentially a method of open-ended questioning with the goal of discerning an individual's problem-solving approach and their operations used in different states of the problem. The second approach, based on sequential discovery (O'Bannon, 1987), was designed to allow the current users to identify specific values and procedural steps related to objects, events, conditions, and activities in their domain of expertise. Information from these two methods can be used to feed into the class hierarchy definitions for the object model and the user interface graphics and tasks. Finally, cognitive network analysis (Snyder, Paap, Lewis, Rotella, Happ, Malcus, and Dyck, 1986) was selected to gain insight into the way these objects and concepts are grouped and related.

The concept discovery method was designed as a two part process, using a structured interview technique and data acquisition sheets based on the sequential discovery method. Concept discovery leads users through an exercise to identify and discover the major concepts and objects of interest in their domain. This is a non-judgmental process, similar to brainstorming, in which one or more individuals can participate at the same time. The users' responses comprise a large set of concepts and activities relevant to their domain. For airbase logistics, for example, some of these entities may include supply depots, tactical fighter wings, individual aircraft, specific LRU's of the aircraft, resupply procedures, emergency repair activities, maintenance shops and activities, use of consumable materials, etc. Concepts identified through this process will be captured on index cards, which are used in the next phase, or sequential discovery. Sequential discovery is used to acquire different value sets and procedures used by the various concepts/activities identified. This method requires showing users the concept cards and having them specify the order by which the concepts/activities appear in a given scenario. This technique is used to capture the critical concepts and problem-solving approaches of the user/analysts while reducing the introduction of interviewer bias into the problem domain.

Cognitive network analysis uses the non-procedural concepts and abstract objects specified in concept discovery as elements which can be labeled and grouped in a hierarchical manner. The set of concepts is notated on index cards, and users are asked to sort through the cards and group them according to related higher-abstraction concepts. They assign labels to these groups (or pick the labels from the concepts). These group labels become the higher level concepts used to derive the cognitive network. Users are then asked to assign a distance rating value to each pair of high-level concepts indicating how closely the pair of concepts are related. This generates a matrix of pairwise distance ratings. The network is derived by using a mathematical algorithm such as Pathfinder (Schvaneveldt, Durso, and Dearholt, 1985), which identifies the network of the concepts and their relationships.
The value of the conceptual model analysis is to generate user interface prototypes that map to the user's conceptual model of the domain, thus supporting an intuitive and understandable command and menu structure and graphical representations. Additionally, this method supports object-oriented data modeling, providing a natural hierarchy and network of object concepts from which to establish classes, object relationships, and inheritance of attributes.

One of the outcomes from using the cognitive model technique is a definitive set of related concepts and actions which can be mapped to the menu lists for user selections. Since the distance ratings applied by logistics analysts to concepts and actions reflect their idea of related functions, these functions can be designed into the menu and command structure of the environment. The menu structure and command/function design can be evaluated for usability by constructing user interface prototypes as described in the following section.

Interactive Prototyping of the User Interface

Rapid prototyping of the IMDE user interface was planned to capture user interface (UI) design concepts early in the development cycle. The prototyping process provides an opportunity for users, domain experts and developers to work as a team in the UI design process.

Rapid prototyping is the process of using interface design tools to quickly construct user interface functions in software which can be presented to users to gain feedback and agreement early in development. Prototyping is an iterative procedure that encourages successive changes to an original UI prototype, refining the design in software until it closely satisfies the specification goals and meets the user or functional expert's acceptance.

The rapid prototyping process facilitates agreement on the UI design well in advance of implementation, providing a visual specification that developers and users can modify and refine. In addition, the prototyping approach allows the user interface to be constructed as independent modules that can be linked to the program modules. Also, the advance design and testing of the UI reduces the risk of inappropriate application design, as the user interface functions are established and available to developers as guidance.

Since a UNIX-based Sun workstation was selected as the platform for IMDE implementation, the Open Windows™ windowing environment was the natural choice for the user interface environment. Open Windows is a widely-employed windowing system built upon the MIT X-Windows (X.11) standard network-based windowing protocol. The Open LOOK™ user interface standard was adopted as the standard for "look and feel", and Open Windows devGuide (Developer's Guide) was selected as the primary UI builder for prototyping the IMDE user interface.

The use of advanced workstation technology and prototyping tools allows IMDE developers to create a direct manipulation UI representing the user's domain of analysis. According to Shneiderman (1987), a direct manipulation interface expresses the following features: "(1) The user sees a continuous representation of the "world of action." The objects of interest and the permissible actions are represented on the screen in a visual format that takes into account the user's knowledge of the task domain and taps the human skill for analogical reasoning, (2) Physical actions ... (e.g., pointing, selecting icons, drawing) replace typed commands. Selecting from a list reduces memorization, keystrokes, and errors, and (3) Actions are rapid, incremental, and reversible. Furthermore, the impact of an action on other objects of interest is shown immediately (the world of action)" (Baroff, Simon, Gilman, and Shneiderman, 1988).

The IMDE system itself is comprised of three major components: a preprocessor, the simulation processor, and postprocessor. The preprocessor will embody the majority of the graphics-based direct manipulation interface, since the simulation is set up and designed by the user in this mode. The menu actions and concepts derived from cognitive network analysis will be designed into the selection criteria for the preprocessor. This analysis is being performed to support interface development for the four levels of IMDE users described in the Introduction. This method will primarily support the User and Analyst level users in easily accessing all the various related tools and commands within the environment using menu lists in the mouse-driven direct manipulation interface. The Developer interface will support the same functions as the Analyst level, but will also allow the more expert Developer to edit and delete existing classes and to create new classes and methods as required to extend and modify the simulation environment. The Developer interface will support higher abstraction capabilities than the Analyst and increased levels of specialization of class hierarchies, allowing users at this level greater capability to zoom out or into the class structures and to perform inspections and diagnoses of simulation events and interactions.
The critical process in interactive prototyping is involving the expert analyst in the actual user interface design. The expert has the most thorough knowledge of the domain "world of action" that must be represented by the interface. It has been noted (Baroff, et al, 1988) that the expert has mastered symbolic tools (such as existing simulation tools) and is likely to have understanding beyond the current state-of-the-art. Therefore, the logistics simulation experts have a high probability of enhancing the design of the UI to match their mental model of operations and to represent the "world of action" in terms they understand well.

Similar to the knowledge acquisition process, this process of designing the UI based on expert input has been termed "interface acquisition". Like the knowledge acquisition processes described above, interface acquisition requires the interaction of an expert with the UI designer. However, this process can also be enhanced using prototyping tools interactively. Working with an existing UI, the experts can observe the "baseline" set of necessary simulation actions, the experts become more aware of the display techniques and graphics available (as demonstrated and integrated by the UI designer), they will begin to reconstruct the interface to better suit the new way of working in the direct manipulation environment.

By engaging the expert analysts in the process of designing the user task dialogues, candidate icons and graphics, menu systems and simulation process selections, they will reveal to the UI designer their view of the "world of action" being modeled in the interface. This process is interactive in that user interface construction and redesign can be made "on the fly" while working with the analysts, using the devGuide interface builder and simulation graphics primitives. As the UI designer reconstructs the interface to better match each successive set of revisions, the experts can observe how close or far the interface conforms to their (evolving) mental model of an effective simulation in their domain. By iterating the user interface design in this way, the partnership of expert analysts and UI designer will construct a highly usable task flow and menu structure that effectively models the analysts' "world of action".

Interfaces for the four levels of users will be based on this methodology. It is expected that the interactive prototyping method will enhance the effectiveness of the User and Analyst level user interfaces. The Developer and Utility level interfaces should not require the same depth of mental model extraction as the "end user" interfaces, however their design will be based upon the interface acquisition performed for the User and Analyst interfaces. Since the Developer and the Utility levels will have some control over future derivations of the model environment, they will also be guided in their future design revisions and additions by the explicit "world of action" representations used at the User and Analyst levels.

The effectiveness of this process will be revealed during the iterative interface design itself, as successive changes improve the overall coherence and usability of the UI. Usability testing evaluations are planned to evaluate the functionality and usability of the interface before the release of IMDE as a functional preliminary prototype. The objective results of the value of this process will be available following these usability evaluations.

**References**


