Architecture for A User Interface Management System

J.J. Thomas

Battelle Pacific Northwest Laboratories, Battelle Boulevard Richland, Washington, WA 99352, USA

ABSTRACT

The term User Interface Management System has been coined to represent a set of software tools for the construction and control of the interaction dialogue between the user and the computational resource. This set of software tools will have a significant impact on the structure of applications in the future. Its advantages include faster and more reliable application development, consistent user interfaces across application domains, and easily modifiable user interfaces. Also, the user interface management systems are an essential experimental tool for research in the human computer interface. This paper describes a brief history of UIMS and an architecture for a specific UIMS to provide an adaptable user interface.

BACKGROUND

The needs and concepts leading to the current UIMS's have been discussed for many years (Foley 1981; Swartout and Bolger 1982; Buxton and Sniderman 1980). These needs along with the current state of technology lead to a team of researchers from Lawrence Berkley Laboratory, Harvard Holmes, Pacific Northwest Laboratory, James Thomas, and GWU (Foley, Feldman, Kamran, Rogers), to focus attention for an 11 month dedicated effort on the construction issues leading towards UIMS systems. The output was an implementation of a simple UIMS. This UIMS could provide consistent interaction styles across two application domains; first, SEEDIS, a geographical information system from Lawrence Berkely Laboratories, and second, ALDS (Thomas and Hall 1983), a tool for the statistical analysis of large data sets. During this effort, several others joined in concert in offering approaches to UIMS's. These included Grif Hamlin of Los Almos Scientific Laboratories, Dave Kasik of Boeing Computer Services, and Bill Buxton from the University of Toranto.

To help establish the concepts within the UIMS, Grif Hamlin and Jim Thomas proposed a workshop on Graphic Input Interaction Techniques to be sponsored by ACM SIGGRAPH. This workshop was held in Seattle on June 2-4, and is documented in a dedicated issue of ACM/SIGGRAPH Computer Graphics, January 1983. At this workshop, the term UIMS was coined and given its identity as a significant research area seen to have a major impact on application developments in the future. The unanimous agreement of the workshop participants indicated the concepts were understood and could provide a focus for further research.
This unanimous agreement also indicated that others had recognized the importance of the underlined concepts of UIMS and had been working in related fields for quite some time. The majority of the time at the GIIT workshop was spent in further defining the concepts of the UIMS. This established a clear understanding and pointed out: (1) the need for continued interdisciplinary research, (2) the lack of tools for the construction and usage of UIMS systems, and (3) a focus on further research areas.

At SIGGRAPH 82, David Rosenthal proposed that Eurographics sponsor a workshop on UIMS. Also at that time, Frank Moses, Army Research Institute and Jim Thomas proposed a workshop on Understanding User Task Interfaces: Styles and Techniques. These two workshops were proposed to concentrate attention on the software aspects of the UIMS and to clarify the interdisciplinary relationships required to better understand interaction methodology.

Since then, research has continued in several areas of UIMS. This paper discusses one area, the architecture of UIMS specifically to allow for an adaptable (dynamically changing) user interface management system. That is, its primary goal is to allow the system to change and adapt its interaction style to the user's needs rather than requiring the user to adapt to the computer's needs.

ARCHITECTURE

The basic task of the UIMS is similar to language translator systems. That is, the input dialogue is translated into a language controlling the application modules. Therefore, it is not surprising to see that the components of a UIMS have similar components to that of traditional language translators (Aho 1979). A key difference between language translation and the UIMS is that most language translation is one way (towards code generation). UIMS, on the other hand, can provide communication at any level within the UIMS from the interaction feedback tied directly to the interaction technique to an application control training set.

Figure 1 illustrates the architecture for the proposed UIMS.

Interaction techniques are the processes for requesting the response and handling of that interaction. Note that an interaction technique involves a prompt, a language input, an input process, as well as an interaction feedback. Any of these four components of an interaction technique may be null but all must be addressed in defining interaction techniques.

An example interaction technique subdivided into four stages would be function key interaction. The prompt may be the back lighting of the key. The language input is the action of pushing the key and the resulting event. The input processing, the identifications of the function key, and the feedback may be the blinking back lighting signaling the selection.

The interaction techniques will be chosen based on several factors. These include devices available, tasks to be completed, user familiarity with techniques, etc. Since these techniques are separate from the application, changing or adapting these interaction techniques to fit the situation is achievable.