

AUTOMATIC GENERATION OF GRAPHICAL USER INTERFACES

Gurminder Singh and Mark Green

Department of Computing Science
University of Alberta
Edmonton, Alberta
Canada T6G 2H1

ABSTRACT

The research reported here is focussed on the issues involved in automatically generating the presentation component of user interfaces. The design and implementation of the presentation component of the University of Alberta User Interface Management System are described. The system is used for automatically generating graphical user interfaces for interactive applications. The system has been designed to keep the other components of the user interface device independent, keep the designer's interest alive in the design process, make the design process less cumbersome, and reduce the burden of programming as far as possible. The results presented in this report are based on the experience gained through implementing a system to generate the presentation component of user interfaces automatically. The presentation component can be viewed as the lexical level of the user interface.

1. Introduction

There has been a growing awareness in software design of the importance of the user. This concern has manifested itself, for example, in analysis of desirable properties of user interfaces [Cheriton76] and in investigations into the user-friendliness of interactive systems. The concept that the user interface can be treated as a separate module within the whole system, and not simply embedded at a range of points through it, is gaining acceptability [Edmonds81]. The effort now is to make user interfaces more interactive, graphic, forgiving, and self-explanatory. But, unfortunately, the construction of good user interfaces even today remains an expensive, time-consuming, and often a frustrating process [Buxton83]. This prompted researchers in human factors to explore the possibility of automatically generating user interfaces and the notion of a User Interface Management System (UIMS). This paper describes a tool for automatically generating graphical user interfaces for interactive programs and explores the issues related to the process.

1.1. What is A User Interface?

The user interface is the part of a system that handles the interaction between the user and the other components of the system. In order to complete a useful task the system accepts inputs and presents outputs through the user interface. As more interactive systems of comparable functionality become available, their success in the market place is based increasingly on ease of use. Bad user interfaces often cause unnecessary loss of productivity and aggravation. Ease of use, not ease of implementation, has become the crucial design consideration.

The basic structure of a user interface does not change radically over a wide range of applications [Green84a]. There are a number of functions that must be performed by most user interfaces. These functions include error detection and recovery, user protocoling, and undo processing. The concepts of a separate user interface module, separate interface designer, and the common features of the user interfaces have lead to the notion of UIMS.

1.2. Automatic Generation of User Interfaces

The fact that the basic structure of a user interface does not change radically over a wide range of programs and that functions like error detection, error recovery, and help are common to almost all user interfaces leads to the idea of automatic generation of user interfaces. The high cost and large turnaround time for hand coded user interfaces provides additional motivation for the idea.

The automatic generation of the user interfaces has the following advantages:

- 1) It reduces the cost of producing user interfaces.
- 2) It provides a much shorter lead time than the hand coding of the interfaces.
- 3) The low cost and short lead time for the production of the user interfaces makes it possible to experiment with new ideas in user interface design.
- 4) Once the user interface generator is debugged completely, the software it generates is more reliable than hand coded software.
- 5) A particular user interface generator may be used to generate a number of user interfaces which are consistent in their over all approach to functions such as error reporting and help. Familiarity with one such user interface can expedite the learning of the others.

1.3. What is a UIMS?

A UIMS is a collection of software tools supporting the design, specification, implementation, and evaluation of user interfaces [Seattle83]. It performs an important role of mediating the interaction between a user and an application; satisfying user requests for application actions, and application requests for data from the user. It thus provides for the application programmer's problem specific skills to be concentrated on the application, and freed from detailed concern with managing the flow of user actions and responses. UIMSs have also been called "Dialogue Management Systems" [Roach82] or "Abstract Interaction Handlers" [Feldman82]. Over the past few years many models of UIMSs have been proposed and implemented [Newman68], [Kasik82], [Guest82], [Buxton83], [Jacob83], [Olson Jr.83].

2. The University of Alberta UIMS

The University of Alberta UIMS [Green85], [Singh85], [Lau85], [Chia85] is based on the Seeheim model of user interfaces discussed in section 2.1. The design and implementation details of the presentation component of the U of A UIMS are described in this paper. Three main notations have been used for specifying the dialogue between the user and computer. These notations are: recursive transition networks, BNF grammars, and events. A system accepting dialogues specified by recursive transition networks is discussed in [Lau85]. Details about an event language and its implementation can be found in [Chia85]. At the present time the implementation of a grammar based notation has not been started. Support for the application interface model is currently under development.

