

An Editing Model for Generating Graphical User Interfaces

by
Dan R. Olsen Jr.

Computer Science Department
Brigham Young University

Abstract

A basic architecture for a User Interface Management System is presented. The problem of updating a display in response to interactive commands is discussed. The basic architecture is then extended to include basic editing and browsing processes on arbitrary data structures. Editing templates are presented as a technique which embodies the entire manipulation process for a particular data structure / data display combination. Such templates in conjunction with the User Interface Management System are able to automatically provide a majority of the code required in an interactive application.

Introduction

Within the graphics community the concept of a User Interface Management System (UIMS) has come into usage [THO83]. Such systems have been developed to overcome the high cost of implementing interactive graphics programs with quality human-computer interfaces. Most such systems have concentrated on the problem of input dialogue management [KAM83, GRE85, JAC83, VAN83]. Having developed three such systems in our laboratory [OLS83, OLS85a, OLS85b, OLS85c], we have become concerned with the problem of data display in an interactive program. In implementing interactive programs we have found that the input dialogue can be programmed in a matter of hours or days, using our tools, but the code to update the display after each modification to the application data structure takes months to implement.

In attacking this display update problem we approached it from the point of view of an intelligent display processor. Our original architecture for an interactive program is shown in Figure 1.

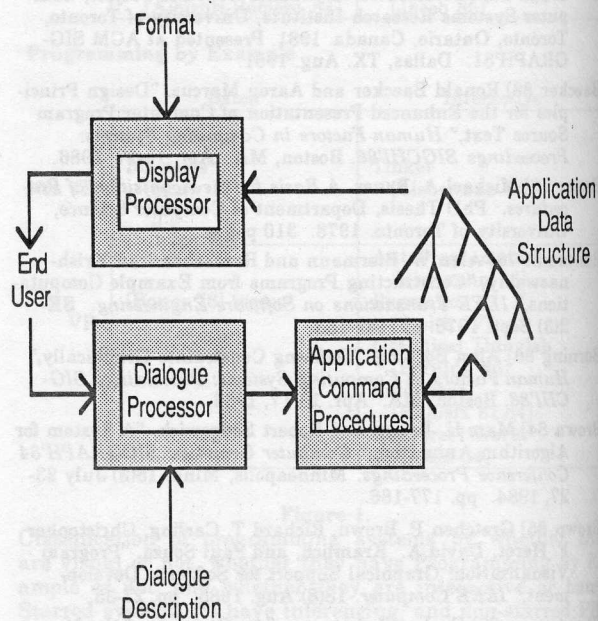


Figure 1.

In this architecture the input events are parsed according to the dialogue description and, based on the input, one or more of the application command procedures is invoked. It is the responsibility of these application command procedures to update some application data structure. It is the role of the display processor to create a graphical presentation of the application data according to the specified format. There are a wide range of possible formats for displaying the data which will only lightly be touched upon here. This architecture is somewhat similar to that proposed at the Seeheim workshop on user interface management [PFA85]. The key problem of interest in this paper is how the graphical image should be updated whenever the application data is changed. The obvious solution is to simply redraw the entire image. This is an extremely poor solution if one desires reasonable response time.

After some experience with the above model we determined that a closer relationship between the data editing commands and the display update functions is essential. A more acceptable system architecture is shown in Figure 2.

