A MODEL FOR GRAPHICS INTERFACE TOOL DEVELOPMENT

Kenneth I. Joy

Signal and Image Processing Laboratory, Division of Computer Science
Department of Electrical and Computer Engineering
University of California, Davis

Abstract

A Problem Solving Environment (PSE) is an integrated system that supports the solution of a given problem, or a set of related problems. Paramount in the development of such environments is the design, specification and integration of user interface tools that communicate between the system and the user. Typically these interactions are object oriented and involve the interaction with tool parameters, which in many applications (CAD/CAM, Imaging Systems, Image Processing), are represented by graphical data. This paper describes a user-interface tool development system in which both textual and graphical display, and interaction techniques are integrated under a single model. This allows the user to interact with tool parameters in either graphical or textual modes, and to have the parameters displayed in the manner most relevant to the problem set. Examples are given of the systems use in two different problem sets.

CR Categories and Subject Descriptors: D.2.2 (Software) Tools and Techniques - User Interfaces ; I.3.4 (Computer Graphics) Graphics Utilities - Software Support ; I.3.6 (Computer Graphics) Methodology and Techniques - Interaction Techniques

Keywords: User Interfaces, Interaction Techniques, Problem Solving Environments.

1. Introduction

A Problem Solving Environment (PSE) [Blinn, 1982; Crow, 1982; Osterweil, 1983; NASA, 1984; DAISY, 1984] configures and manages a set of tools, under a common manager, to support the solution of a given problem. In many scientific/engineering areas, high-level computer software has been developed that allow the user to employ the terminology of the problem area and remove the need to become involved in the design, implementation and maintenance of low level programming code. In these systems, the interface need be configured not toward the design, specification or maintenance of programming tasks, but toward the specification and modification of parameters for tools to be executed, and the management of the execution of the tool. We focus, in this paper, upon a model that allows the integration of a wide variety of user interface tools into a PSE. These tools are used to manipulate the parameters of the application tools of the system.

Interfaces have typically been centered about the command line interpreter [Blinn, 1982; Crow, 1982; DAISY, 1984; NASA, 1984; Osterweil, 1983], or a menu based system [Buxton, 1983]. Some systems have a tutor mode [NASA, 1984] or forms mode [Tsichritzis, 1982], which are based on a limited full screen display. However, display tools in these environments are generally part of the environment tool set and not integrated within the interface manager.

In addition, many of the tools used in scientific and engineering disciplines is based on visual representation of the data. We have found that these representations do not fit well under the command line interpreter, or the menu driven models. In each of these, the display/interaction tools become part of the environment tool set and the user must explicitly invoke tools that display data graphically.

In this paper, we present a user interface model designed to separate the interface tools from the application tool set. These interface tools are integrated into an interface system (see Figure 1), which operates as a companion to a PSE's tool manager. The interface environment now takes over much more of what is currently done in the application tool set: it takes responsibility for the user dialogue, which includes full display and interaction with all data items, and the data management for the application tools. Communication between the managers is done at a high-level [Mamrak, 1982; Liskov, 1977] which allows the user-interface system to work in a more object oriented form.

To achieve integration of different display methods, we base our system about a single abstraction, the ***panel***, which will be our generalized display mechanism. Typically a tool's parameter list is given through a "parameter definition file" (Figure 2 gives an example of a DAISY parameter definition file), and the the data, handled by the interface is generally a set of parameters being passed to a tool. A panel template is used to specify the media in which the data is to be displayed and to give the "display format."

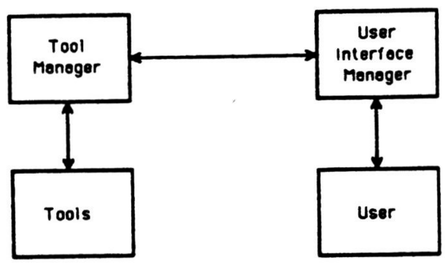

Figure 1: General System Description

```
procedure GEOTRN
 (
      INFILE :      DAISY_FILE ;
                          {Ground Control Point Data}
      OUTFILE :     DAISY_FILE ;
                          {Output Ground Control
                                 Point Data}
      TRANSFILE :         FILE ;
                          {Transformation Coefficient
                                 Data}
      TRANS :       ( bilinear, quadratic ) ;
                          {Transformation Type}
      ACC :         ( standard, extended ) ;
                          {Accuracy}
      THRESH :      REAL
                          {Error Threshold}
 );
      {Generate Geometric Transformation Coefficients}
```

Figure 2 : A Sample Parameter Definition File

The user interface system manages panels. It takes as input a panel template (which specified the format of the data display, and the style of interaction with the data), a parameter definition file and an instance of the data defined by the parameter definition file. It generates the display, which depends on the information in the template, and allows the user to move about and to modify data in the panel. The tool then takes the modified data (if any) and produces a data set that can be passed to an application tool.

The use of a general model, also allows the specifications of panels that not only interact with parameter lists, but can be used to define and interact with high level data types. This allows the user interactions to proceed at a higher level -- dealing with graphical representations of curves, rather than arrays of coordinates.

In this paper, we focus on the user-interface tools and the panel model. The specifics of the interface system can be found elsewhere [Joy, 1984b; Fisher, 1984].

2. The Model

- A *panel* is a display of an instance of a *data type*, corresponding to a given *panel template*.

2.1. The Panel Template

A panel template consists of

- a set of descriptors X_0, X_1, ..., X_n, corresponding to the data items in a given type definition,
- For each descriptor X_i, a display procedure d_i,
- For each descriptor X_i, a (possibly empty) set of user interactions q_1, ..., q_m, and
- for each user interaction specified, a (possibly empty) sequence of tools t_1, ..., t_k that are be executed when the interaction is invoked.

The panel descriptor X_0 contains information specific to the panel as a whole. This descriptor contains the format of the panel, banner information, help information, the positioning of the underlying subdisplays, and history information. If no underlying description is present, the panel layout will default to the textual layout illustrated in Figure 3. Display procedures that correspond to this descriptor have a global affect on the entire panel. Interactions that correspond to this descriptor generally affect the entire panel or concurrent panels representing the same information. Tools associated with this descriptor also act on the entire panel. Figure 7 illustrates a parameter list for a CAD Sweep operation, of which the spine (defined by a B-spline curve) is represented in graphical form.

The descriptors X_1, ..., X_n, each correspond to a single data item from the data set. The information included in the descriptor includes prompt information, help information and local item histories. Interactions with these descriptors generally affect only the individual data items.

The display procedures, d_i, specify how each of the data items are to be displayed to the user. The system has a set of standard items, each of which it "knows" how to display (Integers, Reals, Simple Arrays, Scalar Types, Strings), in which case the d_i will be selected automatically by the system. The model also allows the development and insertion of display procedures for special handling of higher-level data (e.g. coordinate data B-spline curves, surfaces, spheres, etc.). These procedures can be enacted either automatically, or by "selection." All display methods can be overridden, and the default textual data shown.

The tools t are associated with user interactions q on a panel. Each user interaction results in a "standard" operation. For example, the standard operation associated with the modification of a data value on a panel is to modify the data value in the data set (if certain syntax checks are satisfied). If a sequence of tools $\{t_i\}$ is associated with a user interaction q_i then the tools are invoked sequentially after the standard operation has been executed.

If no tools are specified for interaction q_i, the corresponding operation will proceed in a standard manner.

Figure 4 shows a portion of a panel template definition file used to specify shading constants for a Phong [Phong, 1975] shading model. The interaction "modify" is specified for each data item. Upon modification of the data items, the specified routines will be executed.

2.2. The User Interactions

The user interactions apply both globally and locally to a panel. For each interaction, there is a standard operation and a (possibly empty) set of tools.

Global Interactions

- Movement about the panel

  These interactions are used to move the "center of attention" throughout the panel. These operations are determined by the interaction device (e.g. mouse, light pen, or keyboard input) and capabilities of the display. These operations include commands to page through a multi-page panel, to scroll through a single-page panel, or to track a cursor through a graphical display.

- Abort, Undo, History

  The Abort command is used to exit from the panel in a abnormal mode,. UNDO is used to restore the previous version of the data set into the panel display, History is used to review previous panels, or select a previous version of a data set for insertion into the panel display, and Help is used to output help information at the "panel" level.

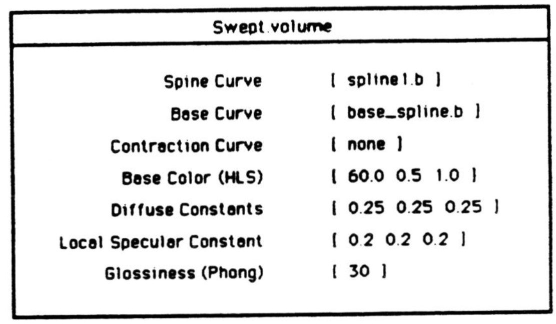

Figure 3: Default Panel Layout

```
/*
  Panel Description for the PHONG shading model
*/

{
  base_color : {
    prompt       : Base Color ;
    type        : 3_tuple ;
    default     : 0.5  0.5  0.5 ;
    modify       : rgb ;
    } ;

  diffuse_constant : {
    prompt       : Diffuse Constants ;
    type        : 3_tuple ;
    default     : 0.4  0.4  0.4 ;
    modify       : rgb ;
    } ;

  local_specular : {
    prompt       : Local Specular Constant ;
    type        : real ;
    default     : 0.05 ;
    modify       : zero_one ;
    } ;

  glossiness_constant : {
    prompt       : Glossiness Constant ;
    type        : real ;
    default     : 40.0 ;
     modify              : positive ;
    } ;

} ;
```

Figure 4 : A Portion of a Panel Template Definition File

• Global Modification

This interaction is takes place whenever any modification takes place on a single data item. The system communicates all changes on one panel, to any other concurrent panels representing the same data. In the Ray Tracing Tool (Figure 6b), the a special display procedure that uses a quick sphere drawing algorithm on an associated frame buffer, executes whenever changes are made to the Sphere panel. In the CAD example (Figure 7), A procedure that redisplays the control points, knot vector and B-spline curve executes whenever changes are made to the B-spline data.

Local Interactions

• Selection

This interaction allows the user to "select" a single item on the screen. This interaction allows the insertion of tools that use alternate display methods (e.g. graphical) for certain data items, special tools for browsing through data sets that are too large to display, or tools that allow the user to select items from a menu. Typically, the selection operation keys special display procedures, for selected data sets. Figure 3 gives the default panel produced by a CAD sweep display tool. The data for the spine is contained in a file. By "selecting" this data item, a procedure that reads in, and expands the data in a graphical form (Figure 7) is executed. This allows the display of, and interaction with, hierarchically defined data sets.

• Entry and Modification of data values.

This interaction is used to modify and enter data values through the panel. The standard operation associated with this interaction is the modification of the displayed data to reflect the full change (some abbreviations can be input, but appear on the display in their full form, once they are validated), and the modification of the data in the corresponding data set. The system automatically does syntax checking for "known" types (integer, real, etc.) and defaults to the attached tools for syntax checking for user-defined types. Modification of graphical panels is accomplished by picking detectable portions of the display, or dragging data items on the display.

• UNDO, History, Help

These interactions corrections correspond to the Global Undo, History and Help, however, they act on individual data items, rather than on the entire display.

2.3. The Interface Tools

Tools are attached to user interactions in the system. Generally, the tools support the entry of specific data types, the examination of data, and execution of special side effects (e.g. display of graphical images on a frame buffer). The current system includes the following tools.

- Array display mechanisms for arrays of any type. Special browser/modification tools for two-dimensional arrays have also been implemented.

- domain checking of parameter modifications

- Special display procedures for graphical representation of data, either through line drawings, or with special rendering routines using a high-resolution frame buffer.

- Communication between concurrent panels representing the same data.

3. Examples

This panel model has been integrated into the DAISY system [DAISY, 1984, Fisher, 1984, Joy, 1984a], a software executive used for Image Processing Applications. This system is used primarily for the display and manipulation of a variety of images. It has one one expanded data type -- the image file.

The GEOTRN panel (Figure 5) is developed from the Geometric transformation tools of the DAISY system. The first three data types are all files (two-dimensional arrays of reals). The Transformation type is a scalar type, as is the Accuracy. Error Threshhold is a real. The first and third parameters can each be selected, at which time they are displayed with an "array modification tool." The system uses pop up menus when modifications of the scalar types is attempted.

The system has also been integrated into DIMAGE - a prototype imaging system developed at UC Davis developed for the testing of CAD/Rendering algorithms. Nearly all data in the system has a graphical representation. The system contains a frame buffer for color images.

The ray tracing tool panel (Figure 6), illustrates the selection mechanism of the system. The user may select and display and of the three items in the data set. The Light Source Information is not contained on a file, but is defined with the parameter list, thus only its type is displayed.

Figure 7 illustrates the CAD sweep display tool panel of DIMAGE. The spine item has been selected. The panel for the B-Spline data type is automatically displayed in both textual and graphical format. The

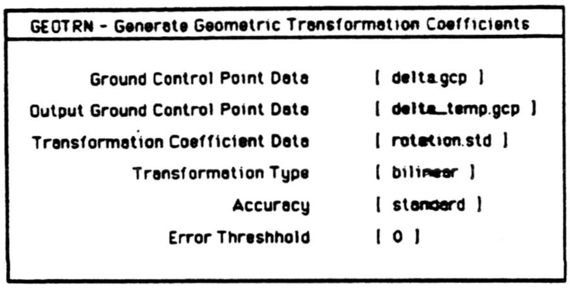

Figure 5: GEOTRN Panel

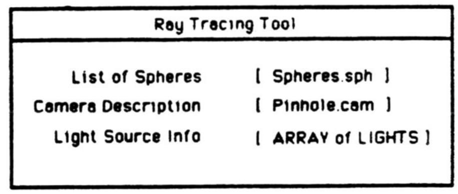

Figure 6a: Ray Tracing Tool Panel

user may manipulate the control points of the graphical display, or may elect to have the spline data displayed textually. The system will all more than one representation of the same data set, and will communicate all changes between the displays.

4. Conclusion

We have described a User-Interface tool management system that is designed to interface with a Problem Solving Environment. This system, constructed under a single underlying display abstraction, the panel, results in an interface environment that is simply defined, simple for the user to control and expandable in its capabilities. It gives a separation of the tools that relate to the display and modification of data, and the application tools that use the data. It allows the system designer to configure a PSE about a number of small tools, and to combine these tools to represent high-level abstractions of data items. The system is capable of displaying and interacting with data through both textual and graphical means, and can display concurrent panels representing the same data in different formats.

Future work with this system includes the further refinement of our graphical format, full integration of the system into a large scale imaging system, and integration of the system into several specialized CAE/CAD environments.

5. Acknowledgements

I thank both the Computer Research Group and Engineering Research Group at Lawrence Livermore National Laboratory for support during the work. Mark Detweiler, Frank Leahy and Jay Govind did most of the programming on the project. This research was supported in part by Lawrence Livermore National Laboratory - Contract 5607805.

6. Bibliography

[Beretta, 1982] Beretta, G. and others, XS-1: An integrated interactive system and its kernel, IEEE Software Engineering Conference Proceedings, 1982, 340-349.

[Blinn, 1982] Blinn, J.F., Systems Aspects of Computer Image Synthesis and Computer Animation, presented at SIGGRAPH 1982, State-of-the-art in Image Synthesis, tutorial notes, 1982.

[Buxton, 1983] Buxton, W., M.R. Lamb, D. Sherman, and K.C. Smith, "Towards a Comprehensive User Interface Management System," *Computer Graphics*, Vol. 17, No. 3, July 1983, 35-42.

[Crow, 1982] Crow, F.C., A More Flexible Image Generation Environment, *Computer Graphics*, 16(3), July 1982, 9-18.

[DAISY, 1983] DAISY User's Manual, Signal and Image Processing Laboratory, SIPL-84-10, University of California, Davis, 1983.

[Fisher, 1984] Fisher, G., DAISY II, A General-Purpose Applications Executive, Computer Science Division Technical Report, CSE-84-2, University of California, Davis, 1984.

[Foley 1982] Foley, J.D., The Design and Implementation of User-Computer Interfaces, Tutorial Notes, Given at ACM SIGGRAPH, 1982.

[Foley, 1981] Foley, J.D., V.L. Wallace and P. Chan, "The Human Factors of Graphic Interaction: Tasks and Techniques," Institute for Information Science and Technology, Department of Electrical Engineering and Computer Science, The George Washington University, Washington, D.C., GWU-IIST-81-3.

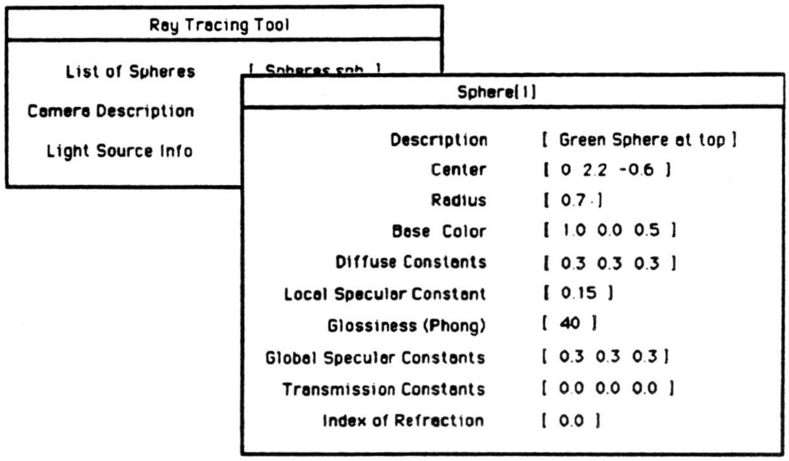

Figure 6b: Expansion of the Sphere Data Item

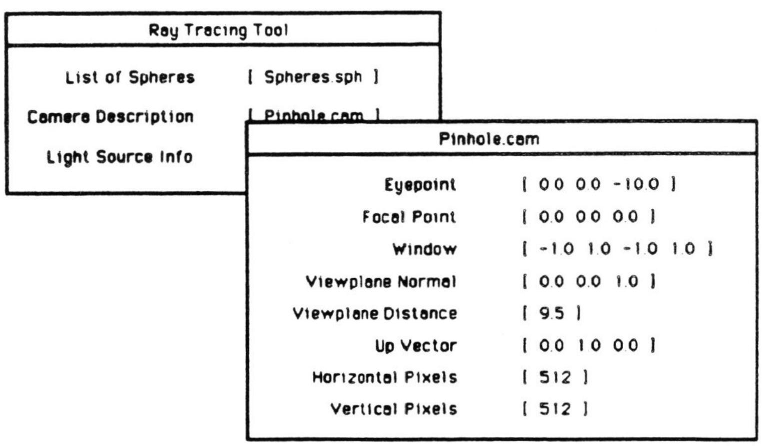

Figure 6c: Expansion of the Camera Data Item

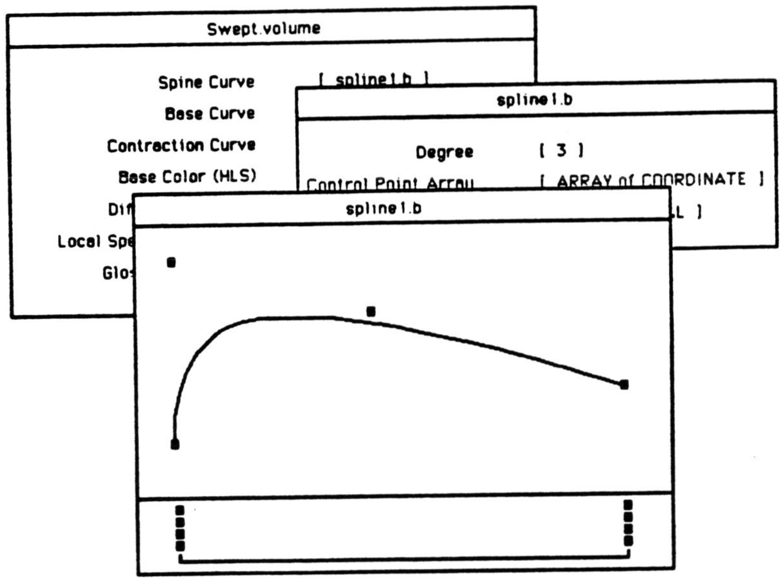

Figure 7: Swept Volume Data Panel

[Goldberg, 1984] Goldberg, A., ***Smalltalk-80 : The Interactive Programming Environment,*** Addison-Wesley Publishing Company, Reading, MA, 1984.

[Joy, 1984a] Joy, K.I., A Model for User Interface Tool Development, Technical Report, CSE-84-4, Computer Science Division, University of California, Davis, 1984, submitted for publication.

[Joy, 1984b] Joy, K.I., Specifics of the DAISY User Interface Model. Technical Report, CSE-84-7, Computer Science Division, University of California, Davis, 1984.

[Joy, 1985] Joy, K.I., Graphics Interface Tool Development in a Problem Solving Environment, Proceedings of Computer Graphics Tokyo, 1985, to appear.

[Liskov, 1977] Liskov, B.H., and S.N. Zilles, "Specification Techniques for Data Abstractions," *IEEE Transactions on Software Engineering,* SE-1, 1(7), 1975.

[Mamrak, 1982] Mamrak, S., R. Dunnington, and B. Shaffer. Installing Existing Tools in a Distributed Processing System, submitted to *ACM Transactions on Computer Systems,* March 1982.

[NASA, 1984] NASA, User's Reference Manual for the Transportable Applications Executive (TAE), NASA 82-TAE-USRVIE, NASA Goddard Space Flight Canter, 1984.

[Osterweil, 1983] Osterweil, L., Toolpack -- An Experimental Software Development Environment Research Project, *IEEE Transactions on Software Engineering,* Vol. SE-9, No. 6, November 1983, 673-685.

[Phong, 1975] Phong, B.T., Illumination for Computer Generated Pictures, *CACM,* 18(6), 1975, 311-317.

[Tsichritzis, 1982] Tsichritzis, D., Form Management, *CACM,* Vol. 25, No. 7, July 1982, 453-477.