APPLIED INTELLIGENCE

OOP's Intuitive Interface Widens Range of Applications

JAMES MARTIN

This is the third in a series of articles on object-oriented techniques, a new technology that's changing the way programmers and users deal with computers.

There are a number of applications today that make use of object-oriented programming. There are varying degrees to which the concepts are used, but in all cases, object-oriented techniques allow the user to deal with a complex environment in a straightforward manner. Some applications simply provide users with an object-oriented interface, while others give users the ability to organize and create objects as well.

This week, we'll look at three examples of object-oriented programming:

- **Window-based user interfaces:** These are the emerging standard for computers, including workstations, Macintoshes and now PCs, with Presentation Manager (PM) and Microsoft Windows.
- **Word processors:** Powerful style sheets let the user create complex document formats with little effort.
- **Graphics programs:** The creation of complex drawings is easier with object-oriented techniques.

The most noticeable effect of object-oriented programming can be seen in the windowing user interfaces that were first popularized on the Macintosh. These techniques are now becoming available on PCs with Microsoft Windows and IBM's PM. The style of interface was first developed for Digitalk Inc.'s Smalltalk, the first pure object-oriented language.

The user benefits from the more intuitive interface provided by object-oriented techniques. The windowing interface is organized like objects in an office. The user can move the objects: open windows, close windows, pull down menus, select items. The effect is the same as opening desk drawers, selecting objects in the drawer, moving a coffee cup and drinking the coffee. Various icons representing objects are used to send messages to the object. The user simply clicks on the icon to obtain the desired result. For example, to scroll a window, the user points the mouse to the scroll bar and clicks it. To close a window, the mouse is clicked in the "go away" box on-screen.

Visual clues are used wherever possible to indicate the behaviors of an object or identify the object. There are icons for the controls on a window and icons used to identify the applications and documents in the system. For example, different icons are used for pictures and text documents. Similar icons and messages are used across multiple applications. Open, close and save operations always appear in the same place on the screen, whether the application is a word processor, draw program or program-development environment. The go-away-box icon is the same for all windows in all applications.

User interfaces based on object-oriented techniques enable users to explore and experiment with a new environment. Many programs can be learned with little or no training. Once users learn the functions of a word processor or program editor, they then can transfer that knowledge to other products that conform to the same user-interface techniques.

Thus, object-oriented interfaces help the user to grasp and work with the complex computer environment. Users learning to operate a windowed interface do so in a fraction of the time it would take them to learn a conventional PC interface, because the windowing techniques help manage and control the complexity of the environment.

One of the more dramatic recent uses of object-oriented techniques is in the definition of style sheets in some word-processing programs. The style sheet defines attributes, such as page layout, font type and text format, for some portion of the document. A complex document might have styles for regular text, bulleted text, headings and subheadings. The style sheets are specified by the user.

Using a conventional approach, the user would fully specify each style sheet. Using object-oriented techniques (where each style is an object), the styles are organized in a hierarchy. The attributes of each style sheet are set at the appropriate level in the hierarchy. Lower-level styles inherit attributes from higher levels, and only definitions that are new or different need to be specified.

For example, the basic text style in a document might define the margins, font and paragraph style. A heading style would be a subclass of the basic text style. It would retain the font and margins, but change the font size and add the bold attribute. A programming book might have a different style for code examples in which only the font is changed. All of these substyles inherit attributes from the basic style, so the user has to specify only what is different for the styles lower in the hierarchy.

This use of object-oriented techniques for style definitions results in a very easy-to-use interface for developing complex documents. The levels of abstraction and inheritance make it easy to work with and control documents that have a complex structure. Again, object-oriented techniques permit the management of complex ideas.

There are numerous programs for producing pictures on a computer, beginning with Apple Computer Inc.'s original MacPaint. Many more of these programs have been developed, and the most sophisticated are using object-oriented techniques to make it easy to create complex drawings.

As shown in the figure, a paint or draw program provides the user with an empty window bordered by a number of icons that represent various drawing tools. These include line-width drawing tools; paint patterns; brush shapes; erasers; tools to draw boxes, circles, polygons; and other special effects. The mouse and a pointer are then used to put the shapes or drawings on the screen. These programs all use the basic object-oriented user interface and employ icons to represent various options.

In contrast, conventional paint programs view the picture as a large bit map. For example, when a box is drawn with a conventional paint program, the bits corresponding to the image are turned on. When a circle is drawn intersecting the box, some more bits are turned on. The program only has "knowledge" of the on and off bits. After the initial drawing, the program has no control over the circle and square and can only add or erase bits. There are copy and duplicate functions, but these work with areas of bits and not with objects in the drawing.

One Shape at a Time

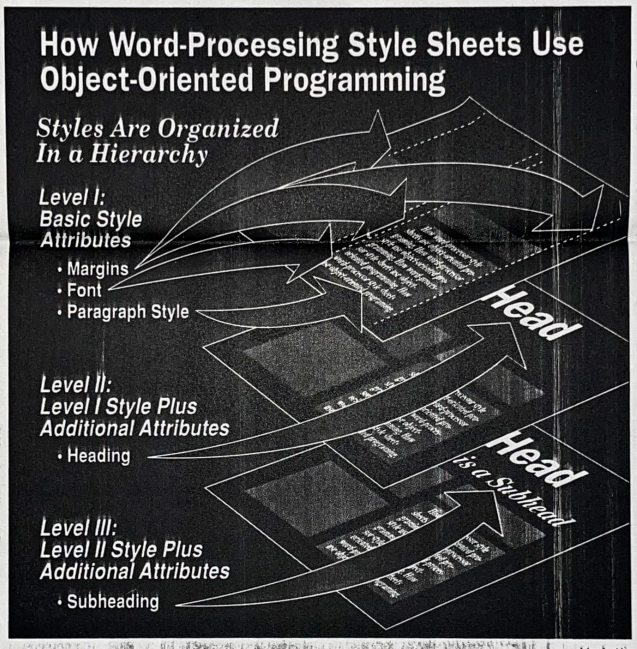
In an object-oriented drawing program, the shapes are remembered by the program as individual objects. They can be selected and modified once drawn. They can be moved about the screen, copied or deleted without disturbing the objects they overlap.

Objects can be combined into more complex objects, which can then be manipulated as single objects. Complex objects can be duplicated and moved, allowing the user to work with the drawing at a higher level of detail.

Each object can be edited. Editing a line means changing its end points. The same edit message can be sent to a rectangle, but with different results: The edit will change both the rectangle's height and width. Each object responds to set attribute messages, but the attributes affect each object differently.

For simple drawings, the conventional bit-mapped paint programs are adequate, but for more complex drawings, the object-oriented approach is recommended. The use of object-oriented techniques simplifies the user interface and helps the user to manage complexity.

Next week, in the fourth and final article in this series, we will look at the growing use of object-oriented techniques to implement CASE tools and databases. ■



John Avakian

Using object-oriented techniques, styles are organized in a hierarchy. The attributes of each style are at the appropriate level. Lower-level styles inherit attributes from higher levels.

To learn more about the subject of these articles, please call The James Martin Report, an information service updated quarterly, at (800) 242-1240. For information on seminars, please contact (in the United States and Canada) Technology Transfer Institute, 741 10th St., Santa Monica, Calif. 90402 (213) 394-8305. In Europe, contact Savant, 2 New St., Carnforth, Lancs., LA5 9BX United Kingdom (0524) 734 505.