

Imperial Software Technology

Visaj

IST first unveiled its Java strategy at the *Software Development West* conference at San Francisco in March 1996. The two products that were announced then — XD/Java and JavaDesigner — are still marketed currently, in the guise of X-Designer Advanced Technology Edition and Visaj respectively.

Visaj was released in October 1997, after a six-month beta test at more than 600 user organizations. X-Designer's good reputation and widespread popularity gave IST a substantial installed base to sell into, and Visaj's family resemblance to X-Designer helped it to catch on. IST goes so far as to state that, because Visaj has proven so easy to learn, it is not planning to offer training courses.

The first question that arises in many people's minds is why anyone would want a specialized GUI builder when there are so many Java IDEs out there. Granted that Visaj complements the standard IDEs rather than attempting to compete with them, it does seem an extra complication — and expense. Looking into the matter a bit further, it turns out that there are at least two good reasons for Visaj's current popularity. In the first place, despite the association of Java with visual applets and JavaBeans, general-purpose IDEs

tend to have rather basic graphical capabilities. Second, more and more of today's development tools — such as Symantec VisualCafé and Sybase PowerJ, and of course Microsoft's Visual J++ — are only available for 32-bit Windows, whereas it is often desirable to develop on Unix. Borland has promised to deliver Solaris and Linux versions of its popular JBuilder IDE, but these are not expected until the end of 1999 at the earliest.

Visaj shows up well on both scores. It runs on Unix and other platforms, as well as Windows, and its similarity to X-Designer helps C++ developers start working with Java. Sun itself has standardized on Visaj for internal use, and another prestigious recent sales win was a leading Internet retailer. IST's CEO Derek Lambert and Sun's Java chief Alan Baratz personally announced the new Visaj WorkShop package, which bundles Visaj with Java WorkShop 3.0. (While Java WorkShop 3.0 supports Java 2, it no longer includes a GUI builder of its own.) TakeFive has also chosen to OEM Visaj, which will be bundled with the Austrian company's highly respected SNIFF+J IDE.

Products

Visaj is pronounced to rhyme with “massage” (i.e., “Veezazh” rather than “Visa Jay”). It is a dedicated

Company Profile

Imperial Software Technology (IST) was founded in 1982 by members of Imperial College, London. It has always specialized in software engineering, and made its name with X-Designer, which became one of the leading Motif interface development tools. Sun's adoption of X-Designer did a lot to boost its popularity, and laid the basis for a partnership that continues currently: IST gets introduced to a lot of new accounts through Sun.

Currently IST is still privately owned, and has its headquarters in Reading, England. Seeking to win sales in the US, the company opened an office in Palo Alto, California in 1995. It is represented in Japan by Marubun Corporation, and in Germany

by Tiscon AG. Elsewhere in the world, prospective customers can contact IST through its Web site — from which they can also download evaluation and licensed software, and obtain support. About 40 staff are employed, mostly on the technical side.

IST's products are used by a wide range of organizations ranging from financial institutions, telecommunications companies and utilities, to defense agencies, jet aircraft manufacturers, research centers, and computer makers. Among its customers are well known names like Bankers Trust, Boeing, BP Oil, CERN, Chevron, the US Departments of Commerce and Defense, Eastman Kodak, EDS, Ford, General Electric, Hasselblad, Hughes Aircraft,

Intel, JPL, Lockheed Martin, Merrill Lynch, Motorola, NASA, Nokia, Northrop Grumman, Oxford University, Pacific Northwest Gas, Saab, Schlumberger, Sony, Sterling Software, TRW, and Xerox.

Imperial Software Technology
120 Hawthorne Avenue
Suite 101
Palo Alto, California 94301
Tel.: (650) 688-0200
Fax: (650) 688-1054
E-mail: sales@ist-inc.com
WWW: <http://www.ist-inc.com>

GUI builder (itself written in 100% Pure Java) that generates Java source code, and thus has to be used in conjunction with general-purpose development tools, such as the JDK.

IST says it has already sold several thousand copies, mostly by way of the Web.

Visaj 2.1

Class Editor

The Class Editor, which appears when Visaj is started, gives a well-balanced overview of the user interface under development. To avoid cluttering the picture with too much detail, subsidiary editors are launched as they are needed to work on specific aspects of the design.

The Class Editor consists of six main elements. Underneath a conventional menu bar and toolbar, it is divided into three panes, each of which can be resized or hidden. On the right is a tree view of the class structure, through which new methods can be created. On the left is a multipurpose pane that normally displays the Beans View including the Object Palette. A tab panel enables the Beans View to be replaced by either the Event Bindings or Method Signature editor.

X-Designer can export bridge files that can be imported into Visaj and used as the basis for Java designs.

Beans View

When the Beans tab is selected, a design area appears between the Object Palette on the left and class tree on the right. User interfaces are built up hierarchically, with components being grouped inside containers, and containers inside a window or dialog. Each container controls the way its children are laid out and how they behave when resized.

Components are selected from the neighboring Object Palette and placed in the design area. Once there, they can be edited and interconnected. Objects are added as children of the currently selected component.

Unlike most other graphical assembly tools, Visaj is distinguished by its use of the more efficient point-and-click technique, rather than the better-known click-and-drag. Simply clicking on a component in the palette places a copy of it on the canvas. This saves a worthwhile amount of time in the course of a day's work, as well as being kinder to the developer's wrist.

Dynamic Display

As soon as the first component is added to the design area, Visaj pops up a floating preview of the GUI under construction as it will appear at runtime. This enables the developer to see the effect of any changes without breaking his or her train of thought, and introducing the delay of code generation and compilation.

Invisible Beans do not appear in the dynamic display, though they are shown in the design area.

Object Palette

Third-party components can be easily imported. IST mentions the following specific sets of JavaBeans as being suitable for use with Visaj:

- *Tea Set Widgets* from InetSoft.
- *JClass Components* from KL Group.
- *Objective Blend* and *Objective Grid* from Stingray (Rogue Wave).
- *Object/FX* from SpatialX.
- *Metamata Suite* from Metamata.
- *JLook* from Loox.
- *Java Foundation Classes (JFC)* from Sun.
- *Application Foundation Classes (AFC)* from Microsoft.

There are two different ways of adding new objects to the palette.

1. *Opening a Visaj palette file* — This affords a finer degree of control over which objects are imported, because objects can be placed in any number of suitably-named palette files. The Swing components can be loaded at startup by launching Visaj with the command: `visaj -swing`

2. *Loading a JAR file* — While simpler, this approach is indiscriminate. All the objects in the selected JAR file are loaded into the palette under a heading copied from the name of the file.

When Visaj starts up, the palette contains a selection of about 30 basic components, together with 16 “Diamonds.” The latter are special Visaj objects.

Diamond Components

Diamonds are special lightweight JavaBeans supplied with Visaj. The following Diamonds are provided:

- *Buttons* — A set of five buttons, ranging from a simple triangle (*ArrowButton*) that can be set to point in any of eight directions, to the *FlexiButton* that can display text, images, or both.
- *FramedPanel* — A single object that draws a frame around a set of related components, with various layouts and appearances.
- *Separator* — A object that can be used to add customizable separators to a visual layout.
- *Controllers* — Sliders, a knob, a meter, and a progress bar.
- *Splitter* — A panel that lays out its children in a grid, separated by “sashes” that can be dragged around within program-defined limits.
- *StatusBar* — A panel that draws a shadow around itself and each of its children.
- *SuperGrid Layout* — A practical compromise between the AWT’s *Grid* and *GridBag* layout managers.
- *DlogTemplateLayout* — A default layout suitable for simple dialogs such as message boxes or error pop-ups.
- *Book (TabbedPanel)* — A tab control component that uses *FlexiButtons* as tabs.
- *ToolBar* — A panel that draws a border around itself. (Docking and drag-and-drop support are planned for future releases.)
- *Bezel Panel* — A container with a bevelled frame, whose edging can be a pixmap.

Project Window

This simple project view has been added to Visaj as an optional way of grouping related files. It is available only when Visaj is run standalone, because its functions are performed by an IDE when one is present.

Resource Bundle Editor

Internationalization is achieved through the use of resource bundles, a generalization of names to multiple languages. Instead of a normal alphanumeric string, a special key is defined. Keys can then be linked to strings in any number of languages by using the Resource Bundle Editor.

This editor displays a table, each column of which corresponds to a language and each line to a key. Instead of the simple English “Yes,” an equivalent key might include the strings “Yes,” “Oui,” “Ja,” “Da,” “Si” etc.

Layout Editors

The layout manager approach is more or less unavoidable when designing pure Java GUIs that are intended for installation on different operating systems. Windows developers are more accustomed to absolute positioning, but this fails miserably when used across multiple environments.

Visaj supports the following layout types by default:

1. *GridBag* (from the Java AWT package).
2. *Grid* (from the Java AWT package).
3. *SuperGrid* (from the IST Diamonds).
4. *Flow* (from the Java AWT package).
5. *Border* (from the Java AWT package).
6. *BoxLayout* (from the Swing component set).
7. *Null* (no layout). This corresponds to absolute positioning.

Layouts are selectable from the palette, and applied to the layout property of the currently selected container.

Event Bindings

All objects in a Visaj design can be linked using events — messages which can be sent by the action of one component and received by the method of another. When a button is pressed, for example, a message is sent to any objects which have registered an interest in this particular event.

The simplest way of creating a new event binding is to wire the source and target objects together by drawing a line between them. This invokes the Event Binding.

The editor shows all the components in the current design, together with all the event types associated with the selected source component and all the methods available for the selected destination component. New events are defined simply by selecting one entry from each column. If the selected method has parameters, these can also be defined.

If a binding is rendered invalid by the subsequent removal of an object, it is marked as such. This is better than automatically removing the binding, because it leaves the developer free to put the object back later.

Properties

All objects have properties, and these can be edited by double-clicking on an object, or from menu and toolbar options. This brings up a property sheet, which lists all the object's properties in a hierarchical tree. A selected property can be modified, provided that it is currently editable; this includes the object's name, scope, and initialization details.

There is also an editor for JavaBeans customers.

Image Editor

Visaj includes its own Image Editor, which is capable of creating both icons and full-size pictures. A full set of features is provided (e.g., pixel placement of text) and any number of images can be worked on simultaneously.

Code Generation

Documented Java source code can be generated at any stage. (It must then be compiled using a third-party product, such as the JDK.) Developers can add extra code if they wish — ideally by subclassing — and this handwritten code is not overwritten when regenerating.

Installation, Documentation, and Help

Due to its tightly focused functionality, Visaj is a relatively compact 4MB download. In fact, there is just one version of the software for all platforms, whether downloaded or installed from CD-ROM, though installation procedures are supplied for all supported operating systems.

Any version of the JDK, from 1.1.3 to 2.0, can be used, though IST recommends at least 1.1.4. Visaj installs painlessly and locates the JDK automatically, without any explicit setting of CLASSPATH or other variables.

The shrink-wrapped kit comes with printed manuals, but online help and documentation are also provided — essential, of course, for those who download the software — in HTML format. There is an excellent tutorial.

Unfortunately, Visaj's Java-based reader is fairly slow and offers only the simplest features, so anyone wishing to print out parts of the documentation, or search it, is better advised to use a standard Web browser. Alternatively, IST now supplies all the documentation in PDF format.

Integration with Third-Party IDEs

Visaj can be integrated with Sun's Java WorkShop. An integration package is available for download from IST's Web site (<http://www.ist-inc.com>).

Visaj can also be integrated with other IDEs, but this may take more work. Information on the subject is posted on IST's Web site.

Visaj Workshop 3.0

Currently in beta, this recently-announced package is expected to be generally available by late October 1999. It consists of Sun's Java WorkShop 3.0, integrated with Visaj and marketed by IST.

IST has also developed a transition tool which helps to migrate GUIs designed with earlier versions of Java WorkShop to JFC. Known as the GUI Migration Tool, it is to be available for free download from both companies' Web sites by the end of June 1999.

X-Designer 5

Originally a Motif GUI builder, X-Designer has accumulated extra functionality over the years. It now supports portability across Motif, Windows, and Java, and it can be used to capture and test sessions in Motif. The latest release, X-Designer 5, also provides Internet connectivity and thin-client partitioning.

X-Designer 5 helps developers translate their Motif and Windows applications to Java by generating appropriate GUI code. Since X-Designer's XD/Capture feature can import existing Motif interfaces without access to their source code, this is clearly a powerful conversion capability.

IST believes that one of the secrets of X-Designer's success is its ability to integrate smoothly with third-party IDEs such as Sun's Visual WorkShop for C++, Centerline's CodeCenter and ObjectCenter, and DataViews. Third-party widgets can also be imported and used from within X-Designer.

Platforms

- *Development:* Visaj is written in pure Java, and it can be installed and used on any platform that supports the JDK. In practice, this currently means 32-bit Windows, Macintosh, Linux, and "any reasonable" Unix.
- *Deployment:* Visaj generates Java source code, which can be incorporated in pure Java applications that run on any platform that supports the JDK.

Pricing

- *Visaj Standard:* from \$495 for the first license.

Opinion Strengths

- Visaj is certified 100% Pure Java, so it can be run on any platform that provides a production-strength Java environment. (Unlike such leading Java IDEs as Symantec VisualCafé and Sybase PowerJ, which are available only on Windows.)
- The containment hierarchy (which is fundamental to Visaj) is a very helpful mechanism for GUI designers. Together with Visaj's powerful layout managers, it offers an approach to screen painting that has significant advantages over absolute positioning.
- The Dynamic Display gives a WYSIWYG picture of the GUI under development at all times, without requiring time-consuming compilation cycles.
- Users praise Visaj's clean, efficient design, intuitive user interface, and ease of use. We have seen no mention of any bugs, and some users have been impressed by the product's stability.
- Java 2 (JDK 1.2) and Swing (JFC) are fully supported, as well as JDK 1.1, AWT, and AFC.
- The Diamond JavaBeans and other components shipped with Visaj offer enough functionality to build quite sophisticated GUIs, even before importing any third-party or in-house components.
- Visaj generates clean, well-formatted, commented Java source code. Handwritten additions are not overwritten when a design is regenerated.
- IST bundles Visaj with Sun's Java WorkShop, and TakeFive also bundles it with its own SNIFF+J.
- The X-Designer import feature lets developers convert their Motif GUIs to Java with a minimum of cost, time, and effort.

Weaknesses

- Not everyone will find it convenient to invest in a separate GUI builder in addition to a Java IDE.
- The downside of Visaj's pure Java implementation is the familiar Java sluggishness when carrying out processor-bound tasks. For instance, it can take a full 10 seconds to move between chapters in the User's Guide.
- The Java-based help and online documentation facility lacks many of the useful features of Windows Help, such as flexible navigation, search, and printing. (This can be worked around by using a standard Web browser instead of the built-in reader.)
- IST's physical presence is limited to the US and the UK, though it has an extensive international distribution network, and products and support can be obtained from its Web site.

Conclusions

There is a lot to be said for choosing a niche and dominating it thoroughly. That is what IST did with the Motif GUI builder space, and it looks well on the way to extending its success to the potentially much larger Java market.

Among the factors in IST's favor are its good reputation in the Motif and wider Unix communities, its close relationship with Sun, and the widely acknowledged quality of Visaj. Within the characteristic limitations of Java, this appears to be a very solid, reliable tool that does exactly what serious programmers expect of it.

On the other hand, we must place in the balance IST's small size, limited resources, privately-held status and relatively recent arrival in the US. One more thing: while Visaj is technically very satisfactory, it attempts to buck the trend toward ever-greater integration and tools that "do everything." (Admittedly, the Visaj WorkShop package is an intelligent move to combat such perceptions.)

Perhaps the greatest threat that IST faces is the sheer size and vigor of the market for Java programming tools. The pace of innovation is punishing, and at any time some other small company anywhere in the world may come up with a competitive product.

Nevertheless, IST is as well placed as it could reasonably hope. If it can strike the right balance between expending resources on product enhancements and marketing, while maintaining its relationship with Sun, Visaj may soon be a standard part of the Java client developer's toolkit.