visualJava, anyone? visualJava, everyone!

Tony Faustini

Department of Computer Science
Arizona State University
Tempe, AZ 85287

E-Mail: faustini@asu.edu
URL: http://www.eas.asu.edu/~lu/visualJava.html

Extended Abstract

visualJava is an entirely visual environment that empowers ordinary internet users to experiment with prefabricated visualJava software components. visualJava is an implementation based on ideas presented in [1,2 and 3]. In visualJava users graphically create live documents (web pages) by selecting prefabricated components, such as textareas, spreadsheets or charting components, and place them directly into a document. As soon as the component appears in the page it is live and can be immediately used. There is no compile/run or design/run mode in this version of visualJava. Liveness means that the user immediately sees the behaviour of the components used in the document. Components that have a document interface interact through a visual language that is created on the visualJava Desktop. For each instance of a component used in a document there corresponds an icon on the visualJava desktop. The icons have a Communication pins on the periphery of their bounding rectangle. The user enables components to communicate with each other by connecting a pin on an icon with a pin on another icon. The communication can be unidirectional or bi-directional depending upon the component's functionality and the pins used to connect one component to another.

There are three types of pin used on visualJava icons

- Input pins
- Output pins
- Input/Output pins

In visualJava it is possible to connect any input pin to any output pin, VJ will not permit an input to be connect to an input or an output to be connected to an output. Pins enable objects to flow from one component to another. VJ is a user language rather than a developer or programmer's language and as such has no explicit notion of type at the user level. If a component does not understand the type of an
object it has been sent it simply ignores the input. An alternative to this approach is to have components generate exceptions, which then have to be explained to the user. visualJava uses a hybrid model of computation based on

- tagged demand-driven dataflow
- data-driven dataflow
- event driven computation

visualJava programs can have feedback loops in them in the same way that operator nets [3]. This is significant because most visual Languages do not permit feedback loops or cycles.

**visualJava Features**

- VisualJava is a Java Applet that empowers users to experiment with and create Java applets without having to write a single line of Java code nor understand anything about objects or types.

- VisualJava is not only for non-programmers. VJ is an extensible open system that enables programmers to write complex and useful visualJava components for use by those who do not have the time to learn Java. Components can be as simple as a textfield or as complex as a visual SQL component (both these components already exist, but maybe you can write a better version).

- Sample Components
  - Buttons, textfield, textareas ....
  - Animation, Sound, imagemaps, URL openers
  - Charting and graphing components
  - Spreadsheet and calculator components
  - Groupware components
  - UNIX like components for grep, sort etc with a simple user friendly interface.
  - Components for accessing Sybase/Oracle/JDBC databases (thank you! Weblogic)
  - Gauges, sliders, dials,...
  - Network components for doing E-mail, telnet etc.

- VisualJava can be used to develop applications interactively. Users create live user interfaces that they can immediately test by visually connecting predefined or user supplied components using an iconic visual language in which they can connect just about anything.

- VisualJava also comes with a number of Java servers that enable groupware and collaborative components to interact over the internet.

- VisualJava incorporates a compiler that automatically generates Java code for distribution as a stand alone applet.

- The visualJava kernel will be CORBA compliant to facilitate the metering and distribution of visualJava components over a "global software network" and the interoperability of visualJava with other component models.

- Since Java is platform and GUI independent so is visualJava.

visualJava is available on the internet at URL http://www.eas.asu.edu/~lu/visualJava.html. visualJava presents the user with a palette of components very much like popular draw or paint programs. The user selects an item from the palette by clicking it. Once a component has been selected the user can place the component onto the web page. This is done by drawing a rectangle like you would in a simple draw or paint program. You simply mouse down on the web page and drag out a rectangle which becomes the place holder for the component that is being embedded into the document. Once a user has embedded a
component into the page an editor for that component automatically appears enabling the user to
customise the component. This is similar to OpenDoc.

In addition to the web page that is being developed visualJava presents the user with another window
called the visualJava desktop. At the same time as the user is selecting the chart or spreadsheet
component from a palette and drawing it in the web page and icon corresponding to that component
appears on the desktop. Each time the user draws an item into the web page a corresponding icon
appears, one for each item placed in the web-page. Although the items on the web page are live they are
alive in isolation. In order for the web page components chart spreadsheet etc to "socialise" or
communicate with each other the user needs to connect the icons together by drawing lines between
icons.

Collaborative Computing or Groupware components

Since visualJava is net-centric it makes sense to have collaborative or groupware components. These are
components that can interact with each other over the internet. Example of these include:-

- A shared white-board
- A spreadsheet with shared cells
- A chat component
- A collaborative 3D sculpting program

Collaborative or groupware components usually have a Java server associated with the service or
sharing. These are implemented as standalone Java applications that reside on the same machine as the
http server for visualJava.

References

   Computer Science Arizona State University, Tempe, AZ 85287, USA
   programming environment. In OOPSLA 88 Conference Proc. San Diego CA (September):
   176-190.