Computer programming in the Internet Age:experiences converting a X-Window monitoring program to Java [1]

Giuseppe Zito

Sezione INFN Bari(Italy)

I report here about the conversion of an online monitoring program with a graphics interface originally developed in Motif to a Java applet. This applet is integrated in an already existing hypertext document used to monitor a subdetector of the Aleph apparatus.The port was easy and most of the problems encountered can be traced to the immaturity of the language.The support from Internet and the platform neutrality concept behind Java,make in the opinion of the author, this language the future of programming.

Key words: Java;Internet;

1 Introduction

Over the past year, the Java programming language has sparked considerable interest among software developers. Its popularity stems from its flexibility, portability, and relative simplicity (compared with other object-oriented programming languages).

Many experiments in HEP are currently evaluating the possibility to use Java in data analysis , visualization and other applications.

In order to have a first-hand experience,I converted to Java a moderately complex application : the Aleph Hcal monitoring display. This program is used to visualize data coming from a subdetector of the Aleph apparatus for monitoring purposes. The original application was written in C and Fortran using X-window for the graphical user interface on Openvms workstations.It is available only on X-terminals. The Java implementation is platform independent and consists in the integration of an applet in an already existing hypertext ("The Hcal Control room") used for monitoring.

2 The user interface of the stand-alone application

The stand-alone application has three main windows always displayed plus other windows that can show up when the user makes some special request. The three main windows are:

- A comand panel wich enables the user to request the display of new events.
- A main window where the event is displayed. This also contains a few menus to make special requests.
- A text window to show a line of data for each event.

Among the facilities available through menu items in the main window,we have:

- The selection of event files through a file selection panel.
- The selection of detector parts to display through a set of buttons each one connected to a detector part or subpart.

[1] Complete document available online with URL http://www.ba.infn.it/˜ zito/chep97pa.html

3 Basic applet structure

The applet uses the following graphics interface classes:

- A class EventDisplay that extends Canvas and contains the event drawing.
- A class CommandPanel extending Frame which contains buttons and menu to drive the event display. This frame floats above the browser window.
- Another frame SelectPanel will appear if you want to select a detector part.When you have finished you can destroy it.
- The main area in the applet window,in addition to the EventDisplay canvas, will contain two buttons that command the apparition of the two frames, and a small scrollable Textarea to show a line of information for each event plus other status information.

4 Threads

We have introduced two threads:

- Thread one is used to draw the event:in this way we are sure that for complex events drawing, the applet wan't get stuck and the commands will still work.
- Thread two is used to get the next event in parallel with event drawing. Events are accumulated in a cache ready to be processed. In this way,the user doesn't have to wait for the next event. Also, if an event is slow to arrive, the user can continue to work with previous events in the cache.

Events are provided from a CGI-script on the computer that is doing data acquisition. The java applet will send the URL of the script with the number of the event written after that separated by the special character "?". The thread will then wait for the "document" returned by this script which is an event ready to be processed (the script extracts the Hcal part of the event and sends only this part).

5 Lessons learned

After spending a three months learning the language, I found the coding in Java easier than the original coding. The whole process has lasted two man-months of work (compared with around a year of the original).

I encountered several problems using Java as a development language and environment. The good news, however, is that these are not inherent problems; almost all stem from Java's relative immaturity. As Java matures, many of these problems should go away.

The development platform was the Java Development Kit (JDK) Version 1.02 under Windows 95 with Netscape Navigator as the target platform. I wrote all of the code by hand using a text editor. I have used exclusively Netscape Navigator 3 to test the applet. I didn't use the Appletviewer since this sometime behaves differently from Netscape.

Up to now, the best platforms to run this applet are PC with Windows95 or NT but presumably this will change in future.The Unix platform has a Java interpreter implementation too slow to be useful.

6 Conclusion

Being in the programming business since 30 years, Java is not my first language. But in a certain sense, it was a completely new experience.The difference was in the incredible support to my work coming from Internet. For example, for the first time I had all the documentation online. For the first time I could look at the source and run programs developed from thousands of other programmers. Problems that previously would have required days to be solved, could be sorted out in few minutes with a query to a search engine like Altavista.

Another important issue brought up by Java is the platform indipendence. To understand the importance of this issue I can report that for this relatively simple program we have already done three upgrades in 10 years because of platform changes! Each time I spent many months doing the upgrade. If the promise of platform independence will be kept from Java, in the future we shouldn't have any more this problem. You write your program in Java and it will run on any platform, present and future. It is especially this concept that is pushing Java among professional developers.

So my conclusion is that,like HTML, Java is not only here to stay but will change completely the way we do our business.