

Experience with Integrating Java with New Technologies: C#, XML and Web Services

Judith Bishop
University of Pretoria
Computer Science Department
Pretoria, South Africa
jbishop@cs.up.ac.za

R Nigel Horspool
University of Victoria
Department of Computer Science
Victoria, Canada
nigelh@uvic.ca

Basil Worrall
University of Pretoria
Computer Science Department
Pretoria, South Africa
bworrall@cs.up.ac.za

ABSTRACT

Java programmers cannot but be aware of Microsoft's new initiative – a complete language, network environment, and a host of supporting technologies under the title of C# .NET. We highlight the advantages of C# by presenting our experiences connecting it to Java in three ways. The first is by providing a platform and language-independent XML-based API called Views for developing programmer-controlled GUIs. Views does not need C#'s resource-intensive Visual Studio development environment, and is also targeted for Unix. The second provides evidence that C# can be linked to Java at the source code level, albeit through C++ wrappers. The third is a means for retaining the useful applet feature of Java in a server-side architecture of .NET's web services. We conclude that many common shared technologies bring Java and C# together and innovative ways of using others will open up opportunities not hitherto imagined.

Categories and Subject Descriptors D.1.m
[Programming Techniques] – miscellaneous.

General Terms Experimentation, Languages

Keywords .NET, C#, Java, platform integration, GUIs

1. INTRODUCTION

In just seven years, Java has made phenomenal inroads into the world of system, business, internet, educational and now scientific programming. Specifically, features which could profitably be used by scientific programmers are cited as object-oriented programming, security both within a program and between programs, parallelism facilities, applets, and access to new resources through class libraries [3].

Like Java, C# co-exists with a runtime environment (like Java's JVM); C# has ways to communicate on the network (like Java's RMI but unlike Java's applets); and several independent technologies are used by both languages (such as XML). Much of C# is the same as Java. What differs is its supporting technologies, and of course all of its APIs (called namespaces).

The full paper is at <http://www.cs.up.ac.za/polelo/interests.html>.

Copyright is held by the author/owner(s).
JGI'02, November 3–5, 2002, Seattle, Washington, USA.
ACM 1-58113-599-8/02/0011.

2. AN INDEPENDENT GUI WITH XML

To provide simple and efficient access to GUIs in C# without using Visual Studio, we designed a customised API, Views [2]. Views uses an XML description to drive the setup of a GUI. The user can subsequently set and get information that appears within the controls. The GUI is realised in C# or Tcl/Tk or Java. Use of XML notation to specify the layout of a GUI is not new: the User Interface Markup Language (UIML) is an XML tagging scheme invented for the same purpose [1]. However, UIML has a complex tagging scheme, while our XML tags are greatly simplified by making them have the same names and attributes as the Winforms controls that they generate.

3. LINKING JAVA TO C# USING JNI

From Java's point of view, C# should appear to be native code as C or C++ is, implying it can be called from Java using JNI, the Java Native Interface. The interface works at the level of method calls, so that native methods can create, update and inspect Java objects and call their methods. The difficulty is that JNI only works for C and C++, not yet for C#. To call a C# method, we wrap it in C++. But C# is managed (i.e. garbage collected) and C++ is not. Fortunately, we can use C++ with what are called Managed Extensions, creating a managed wrapper, which interacts with an unmanaged wrapper, which interacts with Java.

4. .NET WEB SERVICES AND JAVA

Unlike Java, .NET relies on server side processing. There is no equivalent to an applet, where the runtime machine is held in the browser. Although intrinsically more complex than the client-side program model, the .NET environment simplifies the server side processing structure. However, as an alternative to the standard .NET approach we have created and maintained Java applets on the client side that interact with the server as .NET web services through the standard web protocol SOAP (Simple Object Access Protocol). Java supports SOAP through the JAX package.

5. REFERENCES

- [1] R. Cover, User Interface Markup Language, <http://www.oasis-open.org/cover/uiml.html>
- [2] R N Horspool and J M Bishop, Views, <http://www.cs.up.ac.za/~jbishop/rotor>
- [3] C M Pancake and C Lengauer, High-performance Java, Communications of ACM 44 (1) 99-101, October 2001.