BUILDing Dynamic Mobile User Interfaces: Demo Proposal

Svetoslav Ganov, Enos Jones, Angela Dalton, and Dewayne E Perry
Department of Electrical and Computer Engineering
The University of Texas at Austin
{svetoslavganov,enos.jones,adalton}@mail.utexas.edu, perry@ece.utexas.edu

Abstract

Mobile users often work with a variety of heterogeneous devices. Ideally, any one of these could effectively execute their personal applications with a functional user interface appropriate to the device. User interface development should be sufficiently abstract and flexible to allow for use on a diverse set of devices, providing convenience and cost effectiveness to developers and users. We have created a framework, called BUILD (Builder of User Interfaces for Local Devices), for mapping applications developed using an abstract user interface (AUI) library to concrete user interfaces specific to individual devices. Our framework enables on-the-fly modification of the application without the need for source code changes or recompilation. We demonstrate our framework with an application written with the AUI library and executed through mapping to two very different concrete user interfaces.

1. FRAMEWORK OVERVIEW

Our BUILD framework design leverages Java to enable portable application implementation for devices and systems of various capabilities. We focus particularly on making the approach sufficiently generic to accommodate a variety of user interfaces for the same application. The abstract user interface components of the application may be mapped one-to-one, many-to-one, or one-to-many instances to actual user interface components. The same relational mappings may be applied to the method calls for manipulation of these components. For example, abstract text output may map directly to voice output, or a text combo box may map to many voice outputs, each representing selections to be determined using voice recognition.

In addition, BUILD enables changes at run time without need for source code modifications or recompilation. We accomplish this by providing an abstract library and necessary mappings to concrete user interface libraries dependent upon the specific device used by the application.

BUILD uses Java byte code instrumentation, accomplished at class loading time via mappings defined in XML files. By intercepting classes that build the user interface as they are loaded in the Java virtual machine, our framework enables remapping of the Java byte code that makes up the executable portion of the user interface. XML mapping files are supplied by device vendors to specify how the abstract library components should be mapped to the capabilities of the device.

2. DEMO DESCRIPTION

Our demonstration consists of a Java application whose user interface dynamically changes at run time to either a graphical user interface (GUI) or a natural speech user interface (NSUI). The application was originally written using the SWT[1] GUI interface. For the purpose of the demonstration, it was rewritten with our abstract user interface library. Using BUILD, the application can run either mapping to its original SWT GUI interface at run time or to an NSUI. The speech mapping illustrates the interface of the application for a device with only speech output and voice recognition. In our demonstration, we will execute the application on two separate devices to show each concrete UI side by side. The demo will be interactive, providing viewers with a chance to try each concrete user interface. We will also provide an illustration of the process through which the Java byte code instrumentation takes place. The illustration will be on a poster to allow us to talk through the process during the demonstration. Viewers of our demo will also be able to see the actual Java byte code for each version of the application, and the XML mapping files used to accomplish the instrumentation.

Our demonstration shows that the byte code of an application written using our abstract user interface library can be changed at run time to present a user interface as specified by XML file mappings. This is accomplished using our BUILD framework, which substitutes the standard class loader with a custom class loader, instruments class byte code on-the-fly, and renders the specified concrete user interface. Property and XML files allow the BUILD interface layer to intercept abstract classes and methods and change them to the appropriate concrete classes and methods.

3. INFRASTRUCTURE NEEDS

Our demonstration application will be running on two notebook computers, with a third available for illustration of the XML files and source code structure. We will need enough table space for these computers and availability of a power outlet to plug in a power strip, which we will provide.

4. CONCLUSION

We have created a Java application that uses an abstract library to dynamically change its user interface behavior at run time, resulting in two very different user experiences. Additional work is needed to demonstrate the framework on lightweight mobile devices such as Pocket PCs or smart phones which use the J2ME environment. Although there are some differences in Java versions, the byte code instrumentation and XML file mappings are portable to these and other devices as well.

5. REFERENCES

[1] SWT: The Standard Widget Toolkit, 2007. Retrieved October 15, 2007 from Eclipse.org: http://www.eclipse.org/SWT