ESPRIT Project N. 25 338

Work Package H
”User Access”

Survey

ID:	User Access V. 1.0	Date:	28.11.97
Author(s):	A. Sinderman/ E. Triep, Fast e.V.	Status:	
Reviewer(s):		Distribution:	

Change History

Document Code	Change Description	Author	Date
User Access - Survey	First version of document. No changes.	Sinderman E. Triep	28.Nov97

User Access - Stratum

1 Java Enabled Devices

One of the main goals of the Follow Me Project is to provide the user with the access to the system and information through different devices, making use of the most new technologies available in the market without forgetting the traditional user end devices like Fax and Telephone.
Real-time operating system (RTOS) industry standardises on PersonalJava and EmbeddedJava.
The Java enabled devices are emerging like the most spread future technology, but the amount of the actual devices present in the market is still poor.

1.1 Java enabled devices actually placed in the market

Device	Provider	On the market	Reference
Web phone	Alcatel	Test version end '97 Commercially launch 2nd quarter 1998	yannick.waelly@bsf.alcatel.fr http://www.alcatel.com/our_bus/telecom/products/bsd/term/product.htm
PowerTouch screen-based telephones	Nortel	Middle 1998	http://java.sun.com/events/jibe/daytwo.html
Mobil telephone with touch screen	Nortel	Middle 1998	http://java.sun.com/events/jibe/daytwo.html http://java.sun.com/pr/1997/august/pr97826-03.html
Web phone Mobile phone	Samsung	Middle 1998	http://www.samsungelectronics.com/~webadmin/news/indext.html http://java.sun.com/pr/1997/august/pr97826-03.html E-Mail : Betty Kim : Product Planning Team

There are nothing announcing the near presence of PAD and Fax devices Java enabled in the market. The above mentioned companies signed agreements with Sun Microsoft looking for a close co-operation an the stark purpose of constructing Java enabled devices.

1.2 Personal Java

Personal Java is a Java Application Environment (JAE) specifically designed for building network-connectable applications for consumer devices for home, office, and mobile use. It is comprised of the Java virtual Machine and a subset of Java, including core and API's and class libraries. It includes also specific features required by consumer applications in resource-limited environments. Personal Java shares a common core set of API's with EmbeddedJava and has the ability to run applets, support more sophisticated visual displays and connect to networks and supports robust user interfaces. Examples of such devices include mobile hand-held devices, set-top boxes, smart phones, etc.
PersonalJava runs on the top of the Virtual Java Machine, ensuring that code runs across a variety of operating systems and microprocessors.
Writing PersonalJava applications leverages the ease of development, code reuse, and inherits security built ino the Java language.

1.3 Embedded Java

Embedded Java is a new Java application Environment for high volume embedded devices.
It consists of core and standard extension API's, and it is designed specifically for severely resource constrained environments.
EmbeddedJava includes a feature level subset of Java, therefore EmbeddedJava applications are upward compatible to both Personal Java and Java. It is used in devices like mobile phones, networking switches, etc.
The EmbeddedJava API is designed to be easily portable to any real time operating system, and to run on a variety of microprocessors.

1.4 Telephony

The Java Telephony API (JTAPI) is a portable, object-oriented application programming interface for Java-based computer-telephony applications. JTAPI serves a broad audience, from call centre application developers to web page designers. JTAPI supports both first- and third-party telephony application domains. The API is designed to make programming simple applications easy, while providing those features necessary for advanced telephony applications.

2 Information exchange between User Access and Agents

The obvious need to interchange information between agents, who are going to require output and input to/of the user through the User Access, and the User Access, conducted us to the search of a generic notation/language describing data to be deliver, layout of the data to be deliver and the delivery rules to be respected according to each specific device.

2.1 The eXtensible Markup Language: XML and eXtensible Style Language: XSL

XML

The goal of the World Wide Web Consortium (W3C) SGML activity is to enable the delivery of self-describing data structures of arbitrary depth and complexity to applications that require such structures.
XML is a subset of SGML. It retains the key SGML advantages of extensibility, structure, and validation in a language that is designed to be vastly easier to learn, use, and implement than full SGML.
XML differs from HTML in three major respects:

1. Information providers can define new tag and attribute names at will.
2. Document structures can be nested to any level of complexity.
3. Any XML document can contain an optional description of its grammar for use by applications that need to perform structural validation.

XML has been designed for maximum expressive power, maximum teachability, and maximum ease of implementation. The language is not backward-compatible with existing HTML documents, but documents conforming to the W3C HTML 3.2 specification can easily be converted to XML.

The applications that will drive the acceptance of XML are those that cannot be accomplished within the limitations of HTML. These applications can be divided into four broad categories:

1. Applications that require the Web client to mediate between two or more heterogeneous databases.
2. Applications that attempt to distribute a significant proportion of the processing load from the Web server to the Web client.
3. Applications that require the Web client to present different views of the same data to different users.
4. Applications in which intelligent Web agents attempt to tailor information discovery to the needs of individual users.

The alternative to XML for these applications is proprietary code embedded as "script elements" in HTML documents and delivered in conjunction with proprietary browser plug-ins or Java applets. XML derives from a philosophy that data belongs to its creators and that content providers are best served by a data format that does not bind them to particular script languages, authoring tools, and delivery engines but provides a standardised, vendor-independent, level playing field upon which different authoring and delivery tools may freely compete.

XSL

XSL is a stylesheet language designed for the Web community. It provides functionality beyond CSS (e.g. element reordering). It is expected that CSS will be used to display simply-structured XML documents and XSL will be used where more powerful formatting capabilities are required or for formatting highly structured information such as XML structured data or XML documents that contain structured data.

Web authors create content at three different levels of sophistication:

- markup: relies solely on a declarative syntax
- script: additionally uses code ”snippets” for more complex behaviours
- program: uses a full programming language

XSL is intended to be accessible to the ”markup” level user by providing a declarative solution to most data description and rendering requirements. Less common tasks are accommodated through a graceful escape to a familiar scripting environment. This approach is familiar to the Web publishing community as it is modelled after the HTML/JavaScript environment.

The powerful capabilities provided by XSL allow:

- formatting of source elements based on ancestry/descendency, position, and uniqueness
- the creation of formatting constructs including generated text and graphics
- the definition of reusable formatting macros
- writing-direction independent stylesheets
- extensible set of formatting objects

2.2 Java Beans

With the advent of the JavaBeans component standard, Sun Microsystems is finally able to shed Java's image as a top ware applet generator. The power and versatility of JavaBeans presents a good alternative to microsoft's ActiveX for building business applications for the Web.

JavaBeans lets you share and reuse components easily and deploy Java-based components in new ways not possible before.

There are two approaches to building JavaBeans: You can publish what a JavaBean can do simply by informing to certain naming conventions, or you can employ a more advanced method involving descriptor classes, which describe what properties, methods, and events a bean can offer explicitly by filling classes with this information. Either approach will result in an easily reusable JavaBean.

The JavaBeans component model includes a slew of additional features. These include the ability to archive, download, and execute JavaBeans remotely, as well as the ability to save and restore data--referred to as persistence. The JAR file specification lets you download beans to a Web page, then install and run them inside your browser.

Other features include security APIs, which ensure security over the Internet, and Remote Method Invocation (RMI), which lets remote machines invoke beans, thus making distributed computing on Java a possibility.

A JavaBean must conform to certain design rules. It must publish its properties (for instance, visual characteristics like size or colour), its methods (what functions it knows how to provide), and its events actions, such as mouse clicks, that it can pass up to its parent). The JavaBeans specification offers an elegant and simple solution by supporting reflection and introspection, processes that enable a container to read the contents of a bean.

2.3 Swing

Swing is a new set of GUI components that include many new features and capabilities that make them more flexible and more powerful than pre-SwingAWT controls. The new features of Swing controls include a modified MVC (model-view-controller) architecture, a new set of new pluggable look-and-feel capabilities that can be defined at runtime among others.

3 Compilant JVM

Both pilot application of the Follow Me project required enabling Netscape 3.0 users to join the system. This users are not able to run a JVM 1.1, which is a mandatory condition to run a place, i.e. it is mandatory condition to be enabled to join Follow Me.

One of the solutions in order to provide a transparent solution to deal with devices JVM 1.0 and JMV 1.1 the same way is the Java Activator.

Java Activator

The Java Activator software provides enterprises customers with the ability to specify the use of Sun's implementation of the Java Runtime Environment (JRE) in Internet Explorer 3.02 or later, and Netscape Navigator 3.0 or later instead of the browsers default Java virtual machine (JVM). Enterprise developers are free to take advantage of all the features and functionality of the latest JavaTM Development Kit (JDK) release and be assured that they can deploy 100% Pure JavaTM applets on their intranet for users of Windows 95 and Windows NT 4.0 desktops.

4 Conclusions

We have research about the technologies available and see that those described in this document are appropriate as a core set for our package, addressing issues such as flexibility, extensibility, portability, high control of the lay out, and slowly becoming adopted standards.

5 References

Graham Hamilton, Java Beans^{TM} Version 1.01 Sun Microsystems.
http://java.developer.com/pages/Gamelan.javabean.documents.html
http://java.sun.com.beans

Sun Microsystems, http://java.sun.com/pr/1997/august/pr97826-03.html

Sun Microsystems, PersonalJava^{TM}
http://java.sun.com:81/products/personaljava/

Sun Microsystems, EmbeddedJava ^{TM}
Samsung, http://www.samsungelectronics.com/~webadmin/news/indext.html

Jon Bosak, Sun Microsystems, XML, Java and the future of the Web

Tim Bry, Beyond HTML : XML and the Automated Web Processing

W3C, Extensible Markup Language (XML)

Sun Microsystems, Java Activator
http://java.sun.com/products/activator/