SUMMARY ON JAVA TECHNOLOGY AND ITS APPLICATIONS

Shikha Vashist, Ayush Gupta

Abstract— **With the rapid increase on requirement of the cross-platform distributed imaging technology, and the common imaging lib, SUN had provided a complete solution based upon Java platform technology. This paper has analysed the evolution of java imaging technology such as [4]Java AWT and Java 2D and Java Advanced Imaging. Especially, it defines the image operators, core classes and the programming framework for Java Advanced Imaging. At last, the paper gives out two imaging examples including the programming codes and the resulted images, which can be used in projects.**

I. INTRODUCTION

For many years, researchers and educators have been investigating techniques for computer-based and computer-stimulated learning. One such technique is known as the interactive illustration. Briefly it is stated that an interactive illustration is a 2D or 3D structured environment that pedagogically guides the user through a concept or set of concepts to foster exploratory learning. Their scope can range from a small responsive [1]2D diagram to a fully immersive and reactive 3D world. Until recently, interactive illustrations were only feasible on expensive workstations found at universities, precluding wide audiences such as high school students who could benefit from them. However, advances in consumer computing hardware and software technology are reducing the problem, and interactive illustrations can now take benefits of commodity computing platforms.

One of the main thrusts of current interactive illustration research is to explore design issues in the context of the World Wide Web. The Web, though still an emerging technology itself has conventions for graphic design, interaction and navigation that should be followed in order to make effective Web-based illustrations. Another research issue is choosing the appropriate illustration development technology. The most prominent player in this realm is Java.

II. WIRELESS JAVA TECHNOLOGY

An apparatus, method, and means are provided for seamless and optimized interaction between users, devices, and applications located in a network environment. The applications may be [6]Java or Java-like applications, and the devices may be mobile communication devices.

Manuscript received Oct 05, 2014

Shikha Vashist, Student, Dronacharya College of Engineering,Gurgaon, Hr., India

Ayush Gupta, Student, Dronacharya College of Engineering, Gurgaon, Hr., India

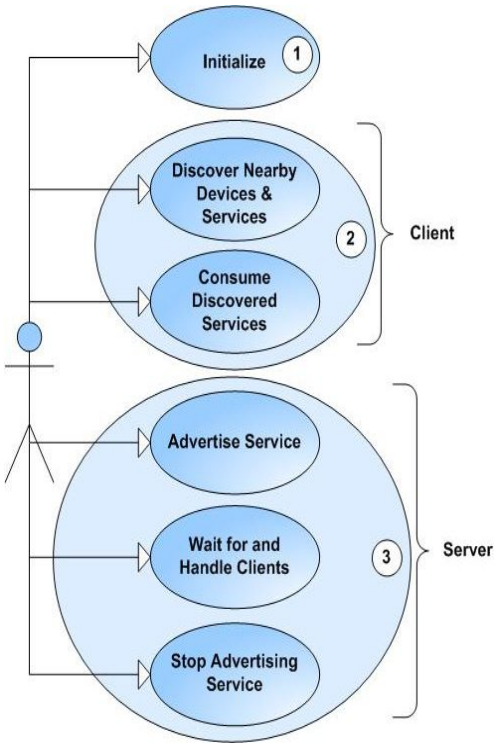

Figure 1

III. ACCESSING AD HOC BLUETOOTH DEVICES FROM A JAVA APPLICATION

Java technology is combined with Bluetooth technology by enabling an access point associated with a Java application to use [2]Bluetooth technology to access data which is stored in a portable Bluetooth communication device and from which a Java object can be produced. The Java application receives a Java object produced from the data, and can then perform a desired function using the Java object

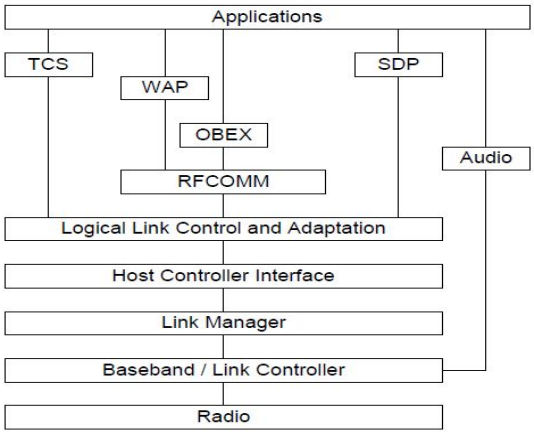

Figure 2

IV. System and method to secure java bytecode code against static and dynamic attacks within hostile execution environments

A method and system that provides secure modules that can address Java platform weaknesses and protect [3]Java bytecode during accomplishment time. The protected modules are implemented in C/C++ as an instance implementation of the protected modules is made in C/C++, this makes use of security technology that protects C/C++ software code.

V. Combo memory design and technology for various tasks as java card, sim-card, bio-passport and bio-id card applications

A combination volatile and non-volatile memory integrated circuit has at least one volatile memory array placed on the substrate and multiple non-volatile memory arrays. The volatile memory and non-volatile memory arrays have address space associated with each other in a way that each array may be addressed with usual addressing signals. The combination of volatile and non-volatile memory integrated circuit further consists of a memory control circuit in communication with external circuitry to obtain address, command and data signals. The memory control circuit depicts the address, command and data signals and transfer to the volatile memory array and the non-volatile memory arrays for reading, writing, programming, and removing the volatile and non-volatile memory arrays. The volatile memory arrays are may be SRAM, a pseudo SRAM or a DRAM. The non-volatile memory arrays maybe masked programmed ROM arrays, NAND assembled flash memory NAND configured EEPROM.

VI. Techniques for buildind and using run-time java archives (jar) for java stored procedures (jsps)

Techniques for constructing and using run-time JAVA Archive (JAR) files for JAVA Stored Procedures (JSPs) are provided. JSP methods composed by an application via a plurality of different JAR files are extracted and packed into a single packed [5]JAR file. When the application tries to execute a JSP, the attempt to process the JSP methods is redirected to a new JSP that calls the single packed JAR file.

Conclusion

Even with its uncertain future, Java is still the best platform for illustration development because of its extensive features and guaranteed audience. The open question is which Java-based technologies are going to prove to be most useful for the development community. The answer will likely be decided within a year's time based on industry support and developer response.

BeanStalkis anexampleofhow currentJavatechnologies canbe appliedtomake the illustration authoring process easier and produce more compelling content. BeanStalk's reliance on AWTis both an asset and a liability. It can take advantage of AWT's numerous features but requires the developer to have a solid understandingofhowAWTworks. Therefore itisnotidealforintroductory students who don't have much experience with object-oriented program design. Visual authoring environments that support JavaBeans may alleviate this problem by abstracting away low-level AWT.

Ultimately; the most important part of interactive illustration development is independent of the technology involved. Experience has shown that attention to graphic design and other user interface issues early in the design process affects the final outcome at least as much as the technology used to implement the illustration. No amount of amazing technology can save a poorly designed interactive illustration.

References

[1]Sun Microsystems, 1997, Aggregation and Delegation Specification (draft). http://splash. javasoft. com/beans/glasgow.html.

[2]Apple Computer, 1997, Agreement Between Microsoft and Apple (press release). http://product.info.apple.com/pr/press.releases/1997/q4/ 970806 .pr.rel.microsoft. html.

[3]Sun Microsystems, 1997, AWT Enhancements in JDK1.l. http://java. sun. com/products/jdk/1.1/docs/guide/awt/designspec/index.html.

[4]Sun Microsystems, 1997, AWT Lightweight UI Framework. http://www.javasoft.com/products/jdk/l.1/docs/guide/awt/designspec/lightweights.html .

[5]J., Academic Press, London, 1997, Beall, J., Doppelt, A. and Hughes, J. "Developing and Interactive Illustration: Using Java and the Web to Make It Worthwhile, "in 3D and the Internet: Information, Images, and Interaction. Edited by Earnshaw, R., and Vince.

[6]Brown University, 1997. Becker, S. Educational Interactive Illustrations. http: / /www.cs.brown.edu/research/graphics/research/illus /thesis/ home .html, Computer Science Honors Thesis.