

Integrating Diverse IT Areas: Bridging the Gap between Multimedia Kiosks and the Wireless World

Stan Kurkovsky, Bhagyavati, Manish Shah

Department of Computer Science

Columbus State University

Columbus, GA, USA

{kurkovsky_stan, bhagyavati, shah_manish}@colstate.edu

Abstract This paper presents a student research effort of developing a multimedia kiosk and extending its functionality to wireless mobile devices. Kiosk Engine (KE) implements a framework for creating and presenting interactive multimedia kiosks. KE is implemented in Java and it can be used to present the kiosk content on both a stationary kiosk and a wireless device. This project brings together many areas of Computer Science: data representation, object-oriented software design and development, wireless applications, and multimedia. We believe that integrating such diverse areas within a single project offers tremendous benefits to the learning experience of students.

Index Terms — Multimedia kiosks, wireless development, mobile devices, Java, UML, XML.

I. INTRODUCTION

Applied research in the major area of study is an integral part of the learning experience of any graduate or undergraduate student. An exposure to modern technologies and tools while still in school gives the students a competitive advantage when they start looking for a job. It is especially important today when many previously plentiful software development jobs in the US have gone overseas because of outsourcing.

Students graduating with computer science degrees must not only know the fundamental principles of their major field of study “in theory,” but they also must be able to apply them in practice. However, it is not enough to train students to be good in developing software. In order to be successful in their profession, computer science students must learn how to use the full spectrum of modern tools, how to quickly adapt to the new development environments and new programming languages, and how to integrate different technologies within a single software project.

In this paper we describe a project undertaken by graduate students at the Department of Computer Science of Columbus State University, GA, USA. This project focuses on developing Kiosk Engine (KE), a portable application for managing content and displaying multimedia kiosk applications. An earlier implementation of this project [8] was presented at the 2003 ACM Mid Southeast Conference and won the 2nd place in the student project competition. Kiosk Engine is written in Java and designed to be portable and lightweight. Kiosk Engine is also extended to support WML and therefore make interactive kiosk content available to all WML-enabled devices, such as mobile

phones with Internet connectivity. In the course of developing of Kiosk Engine, students used the following multifunctional array of tools and technologies:

- **Java** as the primary programming language. **Borland JBuilder Personal Edition** was used as the development environment.
- **XML** as the primary data description language. The structure and layout of the kiosk is encoded using XML.
- **UML** was used throughout the entire project lifecycle to design the Kiosk Engine. We used **Rational Rose** software obtained from Rational Corp. (now acquired by IBM) as a part of a \$500,000 grant.
- **WML** as the kiosk data format for mobile telephones. We used the free **Nokia Mobile Browser Simulator** to simulate the Web browser within the mobile phone. We also used **Apache Tomcat** server to support the **JavaServer Pages** serving the data to **Nokia WAP Gateway**.

Integrating such a large number of modern tools and technologies within a single project provides students with many invaluable opportunities:

- Apply many skills learned in the traditional Computer Science courses;
- Learn how to integrate ideas, skills and technologies from many different areas of Computer Science; and
- Obtain experience in projects with the level of complexity similar to those of industrial projects.

The rest of the paper is organized as follows. Section II discusses the issues typical for the development of multimedia kiosks while Sections III and IV make the case for extending their functionality with wireless capabilities and provide examples of possible applications. Sections V and VI discuss the details of implementing the Kiosk Engine and enabling its wireless functionality. Section VII concludes the paper with a summary.

II. MULTIMEDIA KIOSKS

Interactive multimedia kiosks are typically used to present up-to-date information in a visually appealing format [5]. Historically, kiosks have been created as stationary computers placed in locations convenient to the customers of many organizations [6]. However, with the exponential growth of the popularity of mobile devices, their users too have a need for accessing current information

on the go. Businesses that use kiosks can cater to the mobile user by providing real-time information, which changes dynamically, on the mobile user's handheld wireless device.

Since information on a kiosk can be updated in real-time, it is pulled by the user and not pushed by the business. In today's world of busy travelers and privacy-minded individuals, pull technologies make more business sense than push technologies. Kiosks extended to wireless devices and providing secure access to information can increase the visibility and market share of a business.

Businesses can save time and money by using interactive and custom-made kiosks for presenting their content to users. The customizability of the kiosks ensures their appeal to businesses that are unable to spend resources such as time and labor on making a generic kiosk adapt to their models. Since kiosks display multimedia data, businesses can develop various modes of delivering information. For example, a mobile user who downloads a kiosk application on a handheld device may not appreciate the increased capacity required for viewing videos. Businesses can develop the content of their applications and then present it in various ways, depending on where the application will be executed.

III. WHY EXTEND KIOSKS TO WIRELESS DEVICES?

Because kiosks are interactive, they are attractive to both businesses and end users. Businesses can expect user feedback in real-time and can modify their content accordingly. Users have the satisfaction of dynamic updates of information with interactive capabilities. For example, a static map in a museum provides the users with an idea of where they are, and presents information about the exhibits for possible visitation. However, if an exhibit is temporarily unavailable, the static map is outdated and not of much use. The same museum tour can be incorporated into a kiosk application, which can be used for both a stationary kiosk and also be downloaded by the users into their handheld devices. In this scenario, real-time updates ensure that the information about the temporary unavailability of an exhibit is presented to the user in a timely fashion.

The features of kiosks that make it especially appealing to mobile devices are user interactivity, ease of use and access, and high quality of communication. A kiosk implemented as a lightweight client should be easily adaptable to run on a device with limited processing power to present information in a visually appealing manner to mobile users. The interactive capabilities mean that people on the go do not merely access the content pushed to them, but can also interact and respond with real-time feedback to the provider of the content.

Kiosks on wireless devices can display multimedia data such as images and thus provide high quality communication. Information is presented in a graphical fashion and in an easy-to-use format. Businesses can minimize manual intervention by providing updated content to users on their wireless devices. For example, in a museum, the number of human tour guides can be

minimized by providing the information as a kiosk application on the users' mobile devices.

IV. EXAMPLES OF WIRELESS KIOSK APPLICATIONS

Kiosk applications are becoming popular in today's wireless world [1]. What used to be limited to retail check-out counters is now diversifying into mainstream areas such as banking and tourism [3]. Applications for kiosks abound in the areas of locator services, banking, gaming, Internet access, self-checkout terminals, information access and tourism. As many research reports indicate, there is a growing need for multimodal access to the information presented by kiosks [1,4,7]. Reverting to our museum example earlier, this is an illustration of how a kiosk application is used in the tourism industry. More examples follow.

On a golf course, route information can be provided to users via a kiosk client running on handheld devices. Since route information can change dynamically, a kiosk presents the perfect platform to provide up-to-the-minute information for users. Such information can be useful in preventing users from going to temporarily inaccessible areas. It can also prevent new visitors from getting lost on larger golf courses.

In national parks and animal sanctuaries, visitors can survey route information via mobile devices using a lightweight kiosk application client. Not only will this prevent visitors from losing their way, but it will also help steer more interest into little-known areas of the park. The benefits of providing interactive, real-time content to mobile users can translate to more visitors and higher revenues.

At sport events such as the Super Bowl, fans can easily and conveniently find up-to-date information about sport scores and player statistics. If such a kiosk application were executed on mobile devices, then fans and visitors could keep in touch with each other via their handheld devices, while obtaining the most current information about the sport in real-time. For example, hardcore fans would not want to miss an important event in the game just because they are at the concession counter at that time.

In a ticketing application for a kiosk, people can buy tickets for events via handheld devices. Since clients pull updated, real-time information about events of interest, there is the potential for higher sales volume. In addition, the interactive capabilities of the kiosk imply that busy mobile users can buy tickets after checking the most current information about it on their handhelds.

Kiosks are already being used in the retail industry, such as self-checkout counters in supermarkets. While these applications bring value to the businesses and the users, they can also be provided on handheld and wireless devices to provide additional value for both businesses and end users. Such wireless extensions of existing kiosk applications provide a way for businesses to increase profits while incurring incremental costs for small changes and extensions [2]. Furthermore, a kiosk in a supermarket can allow users to directly go to the aisle with the products of their interest or browse the supermarket catalog pulled by a handheld device in search for a specific product. It saves

shopper time and provides ease of use and access to products.

V. IMPLEMENTATION OF KIOSK ENGINE

We implemented our own Kiosk Engine (KE) as an application that can interpret the description of and present interactive multimedia kiosks differently based on the different inputs provided. Our design and implementation guidelines were as follows:

- It should be easy to create, access (from either web or stand alone), modify and store the kiosk. Consequently, upgrades and changes to a kiosk should be easy.
- KE should be light weight and be able to run on majority of the existing software and hardware platforms.
- KE-driven kiosks should support five basic types of multimedia data: text, images, documents (containing plain/rich text or HTML), audio and video.
- Isolate the data (i.e. text, images, audio, video, etc.) of a kiosk from the processing program.
- The same version of KE should be able to act as a driver for multiple kiosks.

To achieve platform independence, KE is programmed in the latest version of Java 2 programming language with Java Media Framework that provides support for wide range of audio/video formats. The resulting application is truly light-weight – compiled code occupies less than 26 Kbytes. A UML diagram created in Rational Rose showing the class structure of KE implementation in Java is shown in Figure 1.

KE can work as a driver of a kiosk when provided with the input consisting of multimedia data, their description and the overall organization on the kiosk. KE accepts the input written in a specially developed XML-based language called XMLKiosk, which describes the sources, organization, presentation order, size and style of multimedia elements. We chose an XML-based solution for two favorable reasons. The first reason, we are able to create new tags for each basic data type and specify custom attributes for each tag. The second reason, XML can be easily parsed in Java and thus avoid creating a custom

```
<?xml version = "1.0" ?>
<kiosk red="0" green="102" blue="204" start="csumain">
  <screen id="csumain" header="">
    
    </img>
    
      <action type="onclick" alt="Clock Tower" dxpos="344"
        dypos="201" dwidth="13" dheight="37">
        <target>clocktower1</target>
      </action>
    </img>
  </screen>
</kiosk>
```

parser which would add more classes to the code and processing time to the end product.

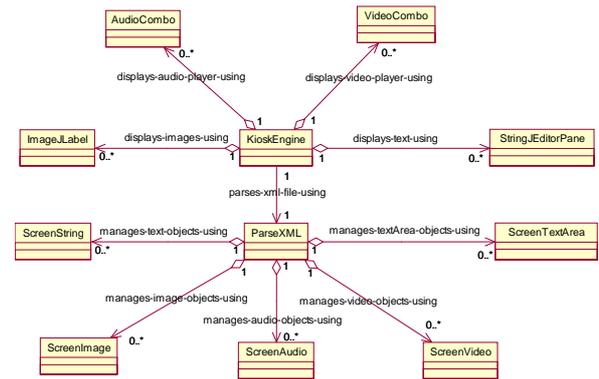


Figure 1. Class diagram of Kiosk Engine.

Each Kiosk application must be described in XMLKiosk with a root XML node <Kiosk>, which contains all the different tags to describe the entire kiosk. Figure 2 presents a fragment of a sample description of a multimedia kiosk, as well as the resulting interactive campus map of Columbus State University (CSU). At the next depth level there are two elements called <header> and <screen>, which are the containers of different types of multimedia objects and both of which can have all or any of the tags for the multimedia objects. To describe the five types of multimedia data namely text, images, documents, audio and video, there are tags called <text>, , <textArea>, <aud> and <vdo> respectively. For each instance of any of the data types on the kiosk screen there is a corresponding tag describing it in the input XMLKiosk file. The <text> and tags occasionally have <action> tag(s), which specify what tasks to perform when some action occurs on the screen due to user's interaction, to facilitate the browsing of kiosk from one screen to another. Also each <aud> or <vdo> tag can have multiple <file> tags which allows a single audio/video instance to play multiple files alternatively.

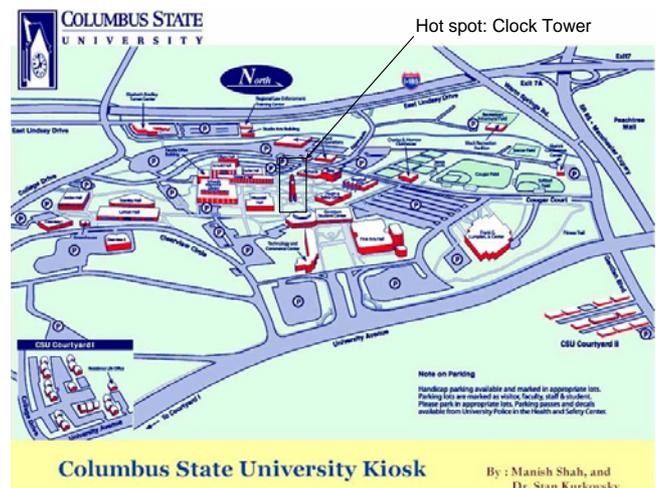


Figure 2. Sample XMLKiosk description of a multimedia kiosk and the corresponding output.

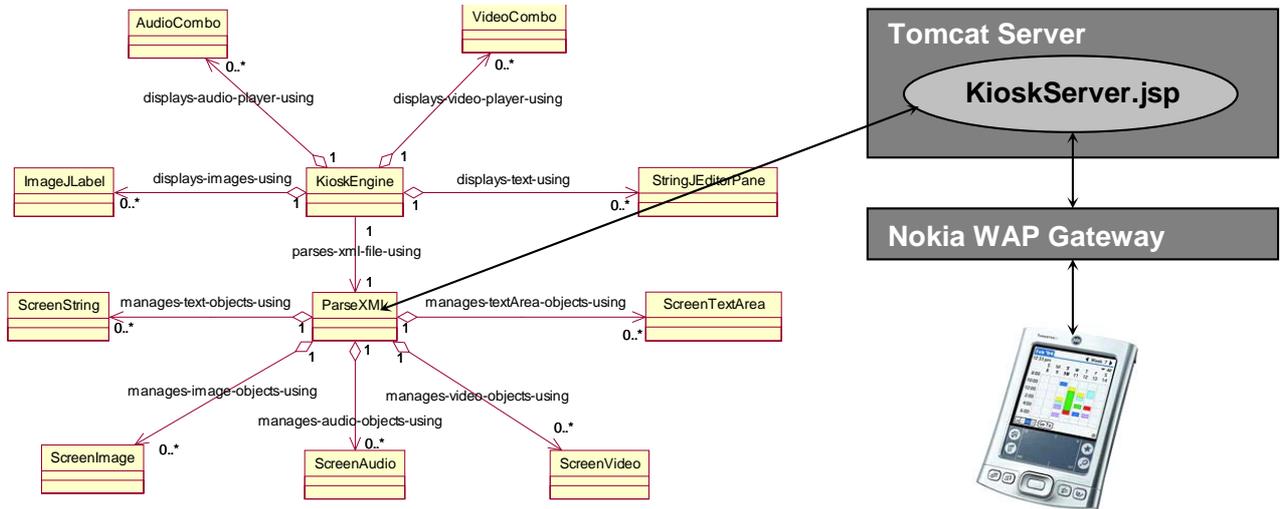


Figure 3. Structure of the wireless extension to Kiosk Engine.

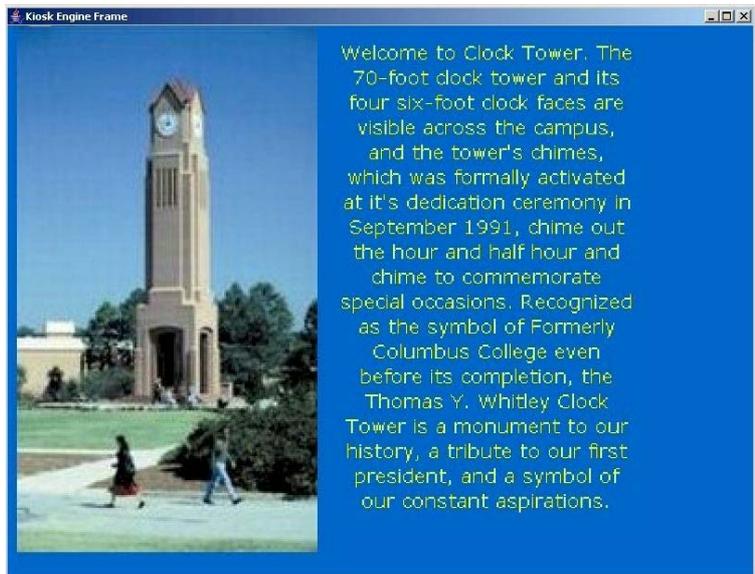
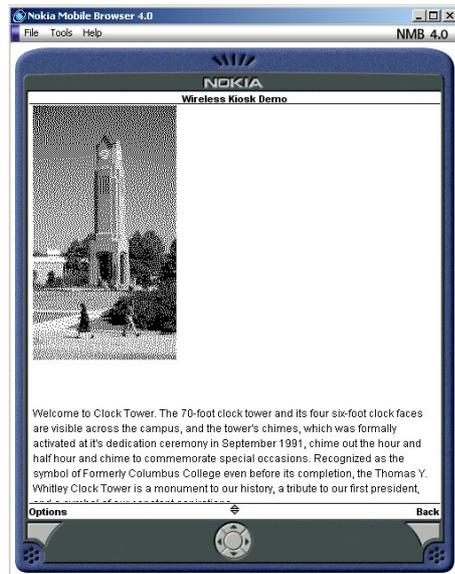
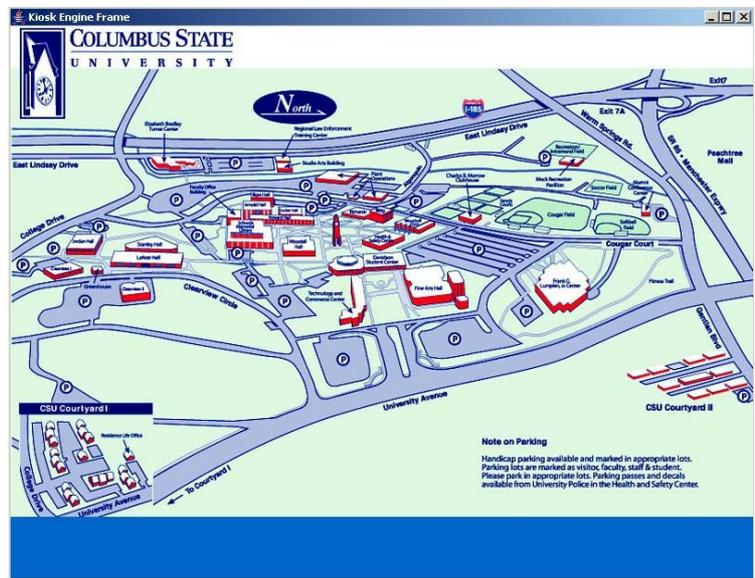
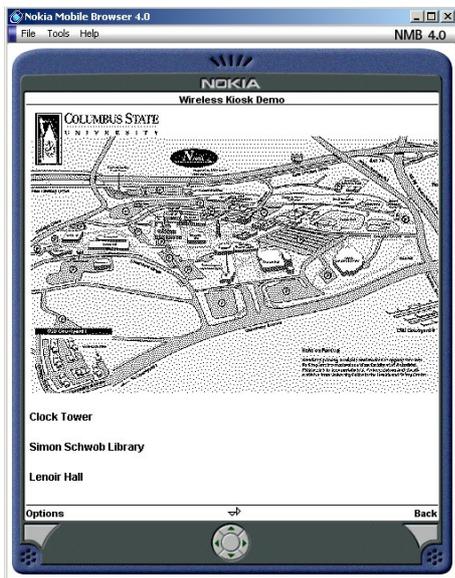


Figure 4. Samples of Kiosk Engine wireless output compared to its standard counterpart

VI. WIRELESS EXTENSION OF KIOSK ENGINE

Having developed a kiosk engine (KE) that placed most of the processing on the server, we decided to extend the KE to wireless devices because of the aforementioned reasons. We used a Wireless Application Protocol (WAP) gateway to convert XML tags to Wireless Markup Language (WML) tags (Figure 3). This gateway or proxy server also converted all image files into WBMP format and performed other conversions in order to be adaptable to a smaller screen size and constrained processing power and storage available on mobile devices.

WML provided the front end and user interface for the kiosk application that employed Java Server Pages (JSP) as the underlying enabling technology (we used Tomcat server). WML tags were dynamically generated based on user input. Since the kiosk is an interactive application, user interface is simple and easy to use. The interface is also visually appealing and invites feedback. We tested the CSU interactive map application on a Nokia simulator. The pilot test ran smoothly and users were able to interact with the application.

The tradeoffs to extending kiosk application on wireless devices consist of limited multimedia content and low-resolution images and graphics, as demonstrated in Figure 4. These limitations are due to the inherent drawbacks of the wireless medium and the intrinsic constraints that mobility imposes on handheld devices. For example, we used no audio or video in our pilot test. The images and maps on a handheld kiosk were grainy when compared to a desktop kiosk. Color images on the wired components were rendered as black and white images on the handheld by the Nokia proxy server, which handled conversion from XML to WML.

VII. SUMMARY

In this paper we presented a student project that involves an extensive integration of many modern software development tools and information technologies. Java, an object-oriented programming language, was used to develop a multifunctional kiosk application with multimedia and wireless capabilities. UML, an object-oriented modeling language, was used throughout the project development lifecycle and supported by the integrated environment of Rational Rose. Nokia and Apache tools were used to support the Internet connectivity and porting of the kiosk data to a mobile phone.

Applying such a widely diverse set of tools provides many advantages to those who use them in an educational setting. Using a combination of these modern technologies provides students with a better understanding of how to apply the skills and knowledge obtained in many different computer science courses within a single project. Such projects will provide a competitive edge to students searching for careers after graduation because of the significant practical software development experience they obtained while still in school.

Authors of this paper hope that their experience of conducting student projects involving a wide variety of

modern software development tools and information technologies will be helpful to others.

REFERENCES

- [1] A. Christian, B. Avery. "Speak out and annoy someone: experience with intelligent kiosks", In *Proceedings of the SIGCHI Conference on Human Factors and Computing Systems*, The Hague, The Netherlands, April 2000.
- [2] R. Denton, P. Jensen. "From VTs to iMacs: moving public computing access into the 21st century", In *Proceedings of the 28th annual ACM SIGUCCS conference on User services: Building the future*, Richmond, VA, October 2000.
- [3] M. Douglas. "Retail kiosks: Breaking new ground", 12 December 2000. Available at <http://www.kioskmarketplace.com/research.htm>; last accessed on 14 December 2003.
- [4] E. Mäkinen, S. Patomäki, R. Raisamo. "Experiences on a multimodal information kiosk with an interactive agent", In *Proceedings of the second Nordic conference on Human-computer interaction*, Aarhus, Denmark, October 2002.
- [5] P. Prasarnphanich and M. Gillenson. "Virtual Extension: The Hybrid Clicks and Bricks Business Model". *Communications of the ACM*, Volume 46, Issue 12, December 2003.
- [6] J. Raymond. "Electronic kiosk project: distributed access to e-mail and web browsing", In *Proceedings of the 28th annual ACM SIGUCCS conference on User services: Building the future*, Richmond, VA, October 2000.
- [7] S. Ross, J. Hill, M. Chen, A. Joseph, D. Culler, E. Brewer. "A composable framework for secure multi-modal access to internet services from Post-PC devices". *Mobile Networks and Applications*, Volume 7 Issue 5, October 2002.
- [8] M. Shah. "Kiosk Engine: A Platform Independent Solution for Multimedia Kiosks," in *Proceedings of The 2003 ACM Mid Southeast Chapter Fall Conference*, Gatlinburg, TN, November 2003.