Integrating Mobile Devices into the Computer Science Curriculum

Qusay H. Mahmoud
Centre for Mobile Education and Research
Department of Computing and Information Science
University of Guelph, Guelph, ON, N1G 2W1 Canada
qmahmoud@uoguelph.ca

Abstract – Mobile devices such as cellular phones and smart personal digital assistants out-ship personal computers (PCs) 20 to 1, and for many students the mobile device is becoming the computer. Such devices are becoming more powerful than the PCs of twenty years ago and they represent a useful tool for conveying important computer science concepts. This calls for innovations in the computer science curriculum, not only in some specific courses but across the curriculum to create a motivating framework for computer science students. After all, students expect faculty to integrate leading edge technology in the classroom. Here we present our approach for integrating mobile devices into the Computer Science curriculum, supported by an example of our experience in integrating BlackBerry devices into two programming courses, a distributed systems course, and senior capstone projects. Some of the courses are lab-intensive where students experiment with the devices, and develop and deploy applications for them. Teaching computer science and programming in the context of mobile applications provides a motivating framework for students and inspires them to excel due to the practical experience they gain allowing them to develop applications for their own mobile devices.

Index Terms – Blackberry devices, CS curriculum, Mobile application development, Mobile devices.

Introduction

With the wide spread use of iPods, cellular phones with multimedia applications, digital cameras and pocket personal computers, there needs to be innovation in the Computer Science (CS) curriculum. Students in introductory programming courses, for example, usually develop and test their programming assignments on a platform similar to the one on which they will be tested by the instructor or teaching assistant. However, this is not the case for mobile applications and services, which are developed on one platform (such as Microsoft Windows or Unix) and deployed and run on a totally different platform (such as a cellular phone). Hence, there exists an opportunity to introduce students to different programming models, but more importantly we believe such experience is practical and inspires students to be excited about learning as it will enable them to develop new applications for their own mobile devices.

The ACM Computing Curricula [1] lists *wireless and mobile computing* as an elective course under the "Net-Centric Computing" body of knowledge, and as a possible consequence few CS departments offer a course on mobile computing at the undergraduate level and those who do the focus of such a course is on theory of wireless local area networks and protocols. The CS curriculum needs to represent today's reality and integrate mobile devices and cover mobile application development. Using mobile devices to assist us in our roles as instructors, and encouraging our students to use their mobile devices to assist them in scheduling tasks among other uses, is valuable and represents an important step towards the integration of mobile devices into the curriculum. But the true integration of mobile devices into the curriculum goes far beyond the role of using mobile devices as personal digital assistants. Students should not be learning about the devices and their features, but rather learning with the devices as they learn concepts and content related to the curriculum, so that integration will contribute to the teaching and learning in the classroom and lab, and this will expand and enhance the learning objectives.

We have integrated mobile devices such as the BlackBerry wireless device into our lower and upper division programming courses, advanced courses, and senior capstone projects. We have achieved excellent results in terms of students excitement and satisfaction, and the feedback we have received from the students in the affected courses is very encouraging and is inspiring us to write about our approach in this paper so that others can benefit from this experience.

In this paper we propose strategies for integrating mobile devices into the CS curriculum, and present our experience in integrating mobile devices into several of our courses through an example of integration. First, we present a brief

This paper was published in the Proceedings of the 38th ASEE/IEEE Frontiers in Education Conference (FIE 2008), Saratoga Springs, NY, USA, pp. S3E-17 to S3E-22.

overview of mobile devices and mobile applications, and the different mobile platforms and learning opportunities they present. We discuss some of the challenges in mobile applications which are practical challenges that provide an inspiring framework for students. We then present our proposed framework for integrating mobile devices across the CS curriculum, and discuss an example of integration that has proven to be successful and beneficial to students in terms of learning and employment opportunities.

Mobile Devices and Applications

Mobile devices out-ship PCs 20 to 1 and as this explosive growth of mobile devices continues, so does the need for compelling mobile applications for such devices. This is especially true given the wide variety of mobile platforms available from different vendors, which has created a competitive market that is good for the consumers, and also triggered a need in a skilled workforce. It is the responsibility of us educators and academic institutions to provide graduates who are capable of designing and developing applications for today's mobile devices as well as the next generation.

Mobile applications are computing entities that can be pushed to users' devices, or downloaded (over the air or through a USB cable) and deployed on such devices [2]. Mobile applications can be categorized as follows:

- *Browser-based applications* developed using a markup language. This is similar to the current desktop browser model where the device is equipped with a browser. For example, the Wireless Application Protocol (WAP), which uses the Wireless Markup Language (WML) or XHMTL for developing applications, complies with this approach.
- *Compiled applications* that require that the device has a runtime environment to execute applications. Highly interactive mobile applications such as computer video games are a good example.

Another appealing stream is the hybrid application model that introduces the best of both worlds. The browser is used to allow users enter the location address or URL to download compiled applications from remote servers, and the runtime environment is used to let these applications run on the device.

Challenges in Mobile Application Development

The explosive growth of mobile devices is stimulating widespread efforts to clone almost any technology developed for desktop computers to mobile devices. Applications such as Facebook and other social network applications are already available on smartphones such as the BlackBerry device. Mobile applications, however, must work within the daunting constraints of mobile devices and this provides CS students with interesting challenges that require brilliant solutions. Consider for example some of the constraints that include memory obligating the developer to consider memory management most carefully when designing application objects for mobile devices. Processing power is another constraint where developers must recognize that elaborate tasks take an unacceptable time to complete. While hardware advancements are making more memory and processing power considerably irrelevant, some constraints such as input and screen sizes do not have hardware solutions. For example, input capabilities are limited and most cellular phones provide only a one-hand physical keypad with twelve buttons: the ten numerals, an asterisk, and a pound sign. In addition, the display might be as small as 96 pixels wide by 54 pixels high. The amount of information you can squeeze into such a tight screen is severely limited [2]. All of these challenges provide a motivating framework for students and inspire them to work hard to design and develop novel solutions that enable them to build mobile applications that at the end of day they can deploy on their inherently personal mobile devices. The end result would be a renewed interest in CS majors and the field in general.

In addition to the limitations imposed by mobile devices, the wireless environment in which such devices operate imposes further constraints, such as the unreliable and expensive wireless networks, the low bandwidth, and the very mobility of mobile devices increases the risk that a connection will be lost or degraded while in the middle of a transaction. Consequence challenges in application development such as transmission errors, message latency, and security also provide a motivating framework for students in advanced courses related to operating systems, networking, and distributed systems.

Mobile Application Platforms

Just like there are several platforms for developing desktop applications, such as MS Windows, Unix, Mac, and others. There are several platforms for mobile devices and mobile application development as well. Some of these platforms and choices for development include: RIM's Blackberry, Microsoft Windows Mobile, Palm OS, Symbian, Qualcomm BREW, Google Android, and Sun Microsystems Java ME [2, 3]. All of these platforms present different technologies for developing mobile applications. Whatever programming language(s) being used in your CS program such as Java, C++, Python to name a few, students can develop compelling mobile applications using any of these languages and deploy such applications on real physical devices.

In our introductory programming courses we use Java and hence we have adopted the Java Platform, Micro Edition (Java ME), in particular the Connected Limited Device Configuration (CLDC) and Mobile Information Device Profile (MIDP), so that students would be able to develop mobile applications without having to learn a new programming language. The Java ME CLDC/MIDP is a standard stack developed through the Java Community Process, and it is available on millions of devices, including BlackBerry devices. The Java ME platform is a viable choice for mobile application development in CS education for a variety of reasons:

- It is based on Java, and it is simpler than Java SE. So if your students are already familiar with Java, they can start developing with Java ME within a few hours.
- Java ME is open source, and it is the most ubiquitous application platform for mobile devices.
- It is widely supported and available on phones from Nokia, Motorola, Sony-Ericsson, and many others.
- There are several open source tools available for developing mobile applications with Java ME.

Software Tools

The software tools needed for integrating mobile devices and mobile application development in the CS curriculum, such as a development environment and an emulation environment, are available free of charge. For example, we have used and recommend using Sun Microsystems Java Wireless Toolkit for CLDC [4], which provides an excellent environment for getting started with Java ME and provides an emulation environment where students would be able to see their applications running in an emulated mobile device environment. Likewise, RIM's BlackBerry Simulator [5] is an excellent tool for emulating the functionality of the BlackBerry device. Our students love the simplicity of these tools.

While running the applications in an emulated or simulated environment provides a good framework for testing such applications, we believe that allowing students to experiment with the physical devices provides a totally different and better experience. In our case, we have provided the students with active BlackBerry devices to experiment with during the labs. In the labs, we have paired the students and provided each pair with a device. The students check out the labeled devices at the start of the lab and return them at the end of the lab. If students happen to have their own Java-enabled devices, we encourage them to use their devices and deploy on them the applications they develop.

Mobile Devices in the CS Curriculum

As mentioned earlier, the ACM Computing Curricula [1] lists *wireless and mobile computing* as an elective course under the "Net-Centric Computing" body of knowledge. In fact very few CS departments offer such courses at the undergraduate level, and some departments offer a variation of this course that focuses on protocols and wireless engineering [6,7]. The CS curriculum needs to reflect today's reality. While offering a course as an elective is advantageous, a large number of students end up graduating without having a chance to take such a course. Therefore, with today's explosive growth of mobile devices, we believe that mobile devices and mobile application development need to be a core topic and part of the undergraduate education.

Topics to be Covered

Mobile devices and mobile application development represent a multidisciplinary area that includes programming, design, software engineering, human computer interaction, Web programming, security, networking, and a number of other traditional computer science areas [8]. Issues surrounding mobile devices and mobile application development must be addressed in courses ranging from CS1 to senior capstone projects.

We are not proposing to turn out mobility specialists with the standard undergraduate programs, but we do feel that students should be introduced to mobile devices and mobile application development not from an end user point of view but from a designer, developer, and software architect point of views. This is best accomplished through an early introduction with continued discussion throughout the curriculum.

As we all know, today every career path open to a CS bachelor's student encompasses aspects of mobile devices and mobility. System administrators need to configure email servers for mobile users; programmers need to build applications to run on mobile devices; web developers need to port an interface for mobile users; and project managers must understand the cost and benefits of porting a system to a mobile device. One could imagine multiple courses on mobile application development being taught, but the topics that need to be addressed include:

- Basic understanding of mobile application development
- Mobile platforms and desired features of future ones
- The mobile device user interface
- Software architecture issues
- Software engineering for mobility
- Networking and security issues
- Databases for mobile applications
- Software adaptability

- Games for mobile devices
- Location-based services
- Service discovery and interaction
- Provisioning of mobile applications and services
- Business models and billing
- Mobile devices in vertical markets such as healthcare

In what follows we discuss integration strategies across the CS curriculum.

Integration Strategies

Introducing new topics or hardware in the curriculum such as mobile application development and mobile devices is challenging for CS departments for a variety of reasons. While creating a new course and adding it to the list of electives is a good start, offering such a course depending on faculty's availability from time to time is not a popular option with students as many of them would graduate without having a chance to take such a course. Here we propose an integration strategy in all appropriate courses across the curriculum, similar to integrating security in CS courses [9]. It is worth noting that this approach is different than the one briefly discussed in [10] for integrating wireless devices into the Information Technology (IT) curriculum in specific courses such as Web development. Mobile devices and mobile application development need to be introduced in early courses and discussion should continue across the curriculum. Here are some guidelines:

- Introduce mobile devices and the basics of mobile application development as early as possible in introductory programming courses. Later we present our example of integrating mobile devices into introductory programming courses.
- Introduce the technology in a lab component in any core CS course.
- Introduce the technology into core CS courses at the intermediate and advanced level, such as data structures, operating systems, database systems; all of these courses provide many opportunities to explore mobile devices and their applications from design and practical point of views. The variety of operating systems available on mobile devices offers a very good use case for an advanced course on operating systems. In a database course students could explore how to access remote databases securely, how to store information in a local database, and how to interact with existing database-driven applications such as personal information management (PIM) data on the device.
- Software engineering and human computer interaction courses have many places in which designing applications and services for mobile devices can be discussed. Many software engineering courses involve a project in which students working in teams design and develop large software systems. Students must understand if their software systems are to support mobile devices, then such design needs to be integrated into the project from the start and not as an add-on.
- Introduce the technology into other project-based courses in network programming, distributed systems, and others. Such courses could be at the second, third, or fourth year levels.
- Introduce the technology in senior capstone projects.

An Example of Integration

In this section we present and discuss an example of integration where we have integrated mobile devices and mobile application development in lower and upper division programming courses, project-based courses, and senior capstone projects. Based on students feedback, we are continuing to integrate this technology into other courses across the curriculum.

Programming Courses

Students learn about the programming models for developing applications for wireless devices and appreciate the unique opportunities such devices offer, but also become aware of the development challenges they present.

We have integrated BlackBerry devices into programming courses (CS1 & CS3) offered in the first and third semesters of study in the Applied Computing program at the University of Guelph-Humber [8]. All of our courses are designed to be lab-intensive in which students learn by doing. The theory students learn in the classroom is reinforced through lab work. We dedicate two weeks of lectures and two weeks of labs for mobile application development towards the end of the semester. Students are introduced to the BlackBerry device and services offered, and the Java platform, Micro Edition (Java ME) CLDC and MIDP. Students learn about the life-cycle of developing applications for such devices. In the first lab students learn how to use the device itself and how to download and deploy existing Java ME applications (known as MIDlets) onto the device using USB cables. In the second lab, students learn how to develop and test their own MIDlets using Sun's Java Wireless Toolkit and its emulation environment, convert the MIDlets into a file format suitable for the BlackBerry device, and then deploy them onto the device.

For the final programming assignment in the course, students are asked to develop two versions of a mortgage calculator application: one for the desktop and another for the BlackBerry device. Students learn that the logic of the application doesn't change no matter on which device it will run. The user interface and interaction model change and thus students learn about the different methods for reading

input from the user and handling events. Figure 1 shows the mortgage calculator application, loaded onto the BlackBerry simulator, developed by students in the introductory programming course.

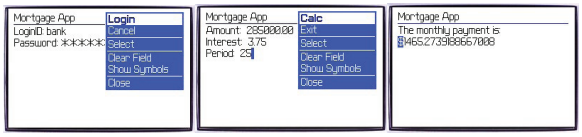

Figure 1: A sample application developed by students in CS1

It is important to note that while we have provided the students with BlackBerry devices to experiment on, the experience they get is not limited to BlackBerry devices. The Java applications they have developed can run on any Java-enabled device and not limited to the BlackBerry device.

In the upper division programming course (CS3), students were asked to develop two applications for the BlackBerry: (1) Apply the Model-View-Controller design pattern to design and develop a multiple-choice quiz that will be marked on the fly as shown in Figure 2, and (2) Develop an application that enables students to retrieve their marks from a remote Web server by interacting with a Perl script. Students have developed these applications successfully, and have enjoyed the experience immensely. This approach has a learning value that you can see on the students' faces. For example, in the upper division programming course one of the first topics covered is threading. Students were introduced to the classical producer-consumer problem and they didn't seem excited about it. However, when shown how threads can be used to deal with user permissions and handling network connections on a real physical device, it all made sense to them.

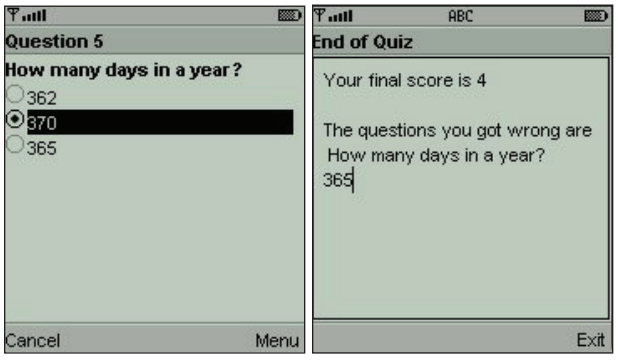

Figure 2: A sample application developed by students in CS3

Developing mobile applications using Java ME is more challenging than developing desktop applications. The Math APIs in Java ME are only a small subset of what's available on the desktop. For example, there is no Math.pow() function and hence students have to either develop this function or be provided with the code for this functionality. In addition, in CLDC 1.0 there is no support for floating point numbers.

Project-based Courses

Several of our courses in the Applied Computing program at the University of Guelph-Humber involve group projects. In the *distributed systems course*, for example, students are asked to work in groups to design and build a distributed system that goes beyond simple message passing. Students were told at the start of the course they can build their systems to support mobile devices, or to run totally on mobile devices, and BlackBerry devices are available for them to test their applications. Several groups have designed and built their applications specifically for Java-enabled devices. One group for example, designed, developed and deployed a version of the Minesweeper game for the BlackBerry device. Their application supports two modes of playing: single player (against the computer), or multi-player in which the players are within a short walking distance of each other where communication between the devices is accomplished through Bluetooth. Figure 3 depicts some screen shots of that game:

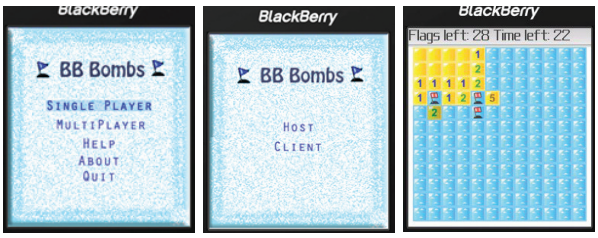

Figure 3: A game developed in a project-based course

Senior Capstone Projects

We have integrated mobile devices into senior capstone projects. Our students spend two semesters working on their final capstone projects. In the seventh semester (students are enrolled in five other courses), students work on a proposal for what they are interested in doing, and in the eighth semester (students are also enrolled in four other courses), students work on implementing their ideas, evaluating them, and writing the final report. Several students elected to work on applications for mobile devices, two of which are described here. One student worked on the design and development of a game known as Hexxagon whose motto is "a minute to learn, a lifetime to master". The end product is shown in Figure 4, and as you can see the student has done a great job in designing a professional and appealing look and feel for this game, which is fully functional and can run on any Java-enabled device. The student has also designed and implemented some novel strategies for playing against the device.

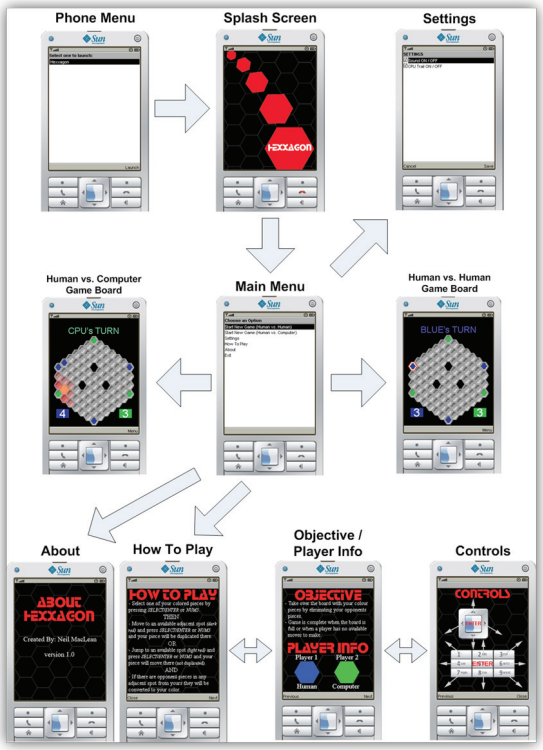

FIGURE 4: A GAME DEVELOPED PART OF A CAPSTONE PROJECT

Another student developed a puzzles suite (Sudoku, crossword, and word search) for the BlackBerry device. For the Sudoku, the student has designed a new algorithm for generating simple, medium, and hard puzzles dynamically on the device. Screenshots of that puzzle suite is shown in Figure 5. This application is fully functional and runs on any BlackBerry device.

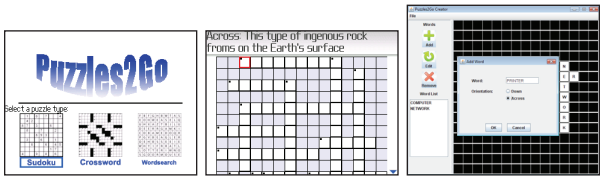

FIGURE 5: A PUZZLES SUITE DEVELOPED PART OF A CAPSTONE PROJECT

Final Thoughts

The integration of mobile devices and mobile application development into our programming courses, the distributed systems course, and senior capstone projects has enabled us to let students work on more interesting and exciting assignments and projects in these courses. This has certainly benefited the students as they have been exposed to a new and rapidly advancing area of computing. In addition, our Applied Computing program has two eight-month co-operative education work placements, and the integration of mobile devices into our computing curriculum has offered our students an advantage in being well prepared for co-operative jobs at software companies that look for expertise in mobile application development. Senior students were able to find full-time employment with such related software companies right after graduation.

When integrating technology into the CS curriculum it is important that the focus be on learning, and hence the need for measuring the learning to show evidence that benefit exceeds the cost. We have used a variety of assessment methods (written surveys, questionnaires, and exist reviews) and we are very pleased with the results. Student feedback has been instrumental in our integration approach. We are designing and building an assessment and evaluation system that can be used by others to help them in measuring the learning outcomes of their students.

Centre for Mobile Education and Research

Providing concrete modules of teaching materials, including lecture notes, labs, assignment suggestions, and tool guidelines would facilitate departments' efforts to include this material in their curricula. We are developing such content and pedagogical methods in the Centre for Mobile Education Research [11] at the University of Guelph. The mission of the Centre is to engage in leading edge applied research to develop state-of-the-art applications and services to facilitate and enhance mobile education and learning, and to provide leadership in integrating mobile devices into the CS curriculum. We are designing and developing academic kits to help universities worldwide to facilitate the integration of mobile devices into CS education. The first academic kit will be available in later 2008.

Acknowldgments

This work and the Centre for Mobile Education and Research are funded in part by Research in Motion (RIM), and I am grateful for their generous support. I would also like to thanks the students in my courses and senior capstone projects for their enthusiastic support of my approach, and for their hard work in developing creative mobile applications some of which are featured in this paper.

Conclusions and Future Work

Integrating mobile devices and mobile application development into programming courses raises the level of excitement and satisfaction and provides a motivation framework for students and the end result is a renewed interest in majors in Computer Science and the field in general. Introducing topics and projects related to mobile

devices and mobile application development does not come at the expense of other topics. They can be used to emphasize the concepts being covered and the related projects would serve only to enliven and motivate the students. This integration however requires commitment on the part of the faculty, as it involves research, development, and a certain amount of creativity. However, it is our responsibility as educators to make sure our students are prepared for the emerging challenges imposed by mobile devices and the need for compelling mobile applications.

For future work, we will continue to integrate mobile devices and mobile application development in our courses across the CS curriculum. In addition to experience reports, we will be releasing an academic kit that others may use to integrate this technology into their courses. It is important to focus on the learning outcomes, and for this we are working on a system for assessing and evaluating the students' learning, which we will be releasing and sharing with the community once it is ready. Please check the website for the Centre for Mobile Education and Research for updates.

References

[1] ACM Computing Curricula (ACM-CC 2001): http://www.acm.org/education/education/education/curric_vols/cc2001.pdf. Accessed: February 7, 2008.

[2] Mahmoud, Q.H., and Maamar, Z., "Engineering Wireless Mobile Applications", Journal of Information Technology and Web Engineering, Vol. 1, No. 1, January 2006, pp. 59-75.

[3] Java Platform, Micro Edition (Java ME): http://java.sun.com/javame. Accessed: January 7, 2008.

[4] Sun Java Wireless Toolkit for CLDC: http://java.sun.com/products/sjwtoolkit. Accessed: January 7, 2008.

[5] RIM's BlackBerry Simulator: http://na.blackberry.com/eng/developers/downloads/simulators.jsp. Accessed: January 8, 2008.

[6] Hu, F., and Teredesai, A., "A Pervasive Computing Curriculum for Engineering and Science Students", IEEE Pervasive Computing, Vol. 6, No. 1, January-March 2007, pp. 88-91.

[7] Bhagyavati, McQueen, T.F., Fleck, R.A., "The Computer Science Curriculum: Make Room for Wireless", Journal of Computing Sciences in Colleges, Vol. 20, No. 2, December 2004, pp. 240-246.

[8] Mahmoud, Q.H., and Dyer, A., "Integrating BlackBerry Wireless Devices into Computing Programming and Literacy Courses", Proceedings of the 45th Annual Southeast Conference, Winston-Salem, NC, USA, March 2007, pp. 495-500.

[9] Null, L., "Integrating Security across the Computer Science Curriculum", Journal of Computing Sciences in Colleges, Vol. 19, No. 5, May 2004, pp. 170-178.

[10] Donnell, P. L., "Integrating Wireless Devices into IT Curriculum", Proceedings of the 6th Conference on Information Technology Education, Newark, NJ, USA, October 2005, pp. 51-55.

[11] Centre for Mobile Education and Research: http://cmer.cis.uoguelph.ca. Accessed: February 27, 2008.