Bringing J2ME Industry Practice into the Undergraduate Classroom

Project Summary

This project aims to advance the state of education for wireless system software developers by adapting and implementing wireless industry best practices in the classroom. Nokia is the leading manufacturer of wireless handsets worldwide. We propose to adapt and implement a software quality assurance program – Nokia's OK Java 2 Platform Micro Edition (J2ME) application certification process – in a core set of wireless software engineering courses. Auburn University initiated the nation's first undergraduate wireless engineering degree program in August, 2002. The courses to be discussed in this proposal make up the software engineering track of the wireless engineering degree.

The objectives of our project are 1) to adapt *Nokia OK* application standards for classroom use as examples of what is expected of industry-quality software; 2) to incorporate *Nokia OK* software development tools into a student-friendly software development environment; and 3) to implement in-house *Nokia OK* certification procedures as a basis for assessing student projects and providing feedback.

Our anticipated outcomes are 1) students who will have been immersed in a technology similar to what they are likely to encounter in their first years of work; 2) wireless industry standards adapted to the classroom, J2ME development capabilities incorporated into an existing popular software tool, and an industry testing process implemented in an academic setting; and 3) evidence of the effectiveness of using wireless industry practices in the classroom.

TABLE OF CONTENTS

For font size and page formatting specifications, see GPG section II.C.

*Proposers may select any numbering mechanism for the proposal. The entire proposal however, must be paginated. Complete both columns only if the proposal is numbered consecutively.

Results from Prior NSF Support

Investigator Cross was awarded NSF grant 9806777 (see [Cross et al 1998]) to investigate the use of visualization techniques in improving the comprehension of software. Although the research was not funded in direct support of undergraduate education, the educational community continues to benefit from the results. The principle artifact of the research was *jGRASP*, an interactive environment available at no cost for developing Ada, Java, C, C++, and VHDL applications. Because of its ability to allow the user to view source code in a variety of visual perspectives, *jGRASP* is particularly useful in teaching software development principles to undergraduate students. *jGRASP* is used in over 100 universities, as evidenced by the jGRASP download log (70,000+ copies downloaded by 48,000+ users), correspondence with users, and the adoption list for the two textbooks which include jGRASP on CD. Publications that acknowledge the NSF award are [Cross and Barowski 2002], [Cross et al 1999], [Cross et al 2001], [Cross et al 2002a], [Cross et al 2002b], [Cross et al 2002c], [Hendrix et al 1998a], [Hendrix et al 1998b], [Hendrix et al 1998c], [Hendrix et al 2002], [Mathias et al 1999].

Goals and Objectives

This project aims to advance the state of education for wireless system software developers by adapting and implementing wireless industry best practices in the classroom. Specifically, we propose to adapt and implement a software quality assurance program – namely, Nokia's OK Java 2 Platform Micro Edition (J2ME) application certification process – in our wireless software engineering courses.

Nokia OK consists of three components: J2ME application standards and style guides; software development and test tools; and certification procedures. The objectives of our proposed project are:

- To adapt Nokia OK application standards for classroom use as examples of what is expected of industry-quality software.
- To incorporate Nokia OK software development tools into a student-friendly software development environment.
- To implement in-house Nokia OK certification procedures as a basis for assessing student projects and providing feedback.

Detailed Project Plan

Context of the project

In response to global trends in wireless communications, one of the strategic goals of Auburn University is to prepare students for meaningful careers in the wireless device industry. To this end, the Department of Computer Science and Software Engineering and the Department of Electrical and Computer Engineering established an undergraduate degree in Wireless Engineering [WEREC 2001]. Students majoring in the degree program take a set of common courses related to wireless technology, and then specialize in wireless hardware engineering or wireless software engineering topics. This proposal describes adapting and implementing industry practice for the core series of software engineering courses used predominantly in the wireless software track.

The courses in the software track expose students to a wide variety of programming languages, devices, and practices, but several core technologies comprise the primary building blocks of the wireless curriculum. These components are the foundational technologies for articulating key concepts and are common threads across the required courses in the major.

Cellular phones are our first core building block. The choice to use cellular phones in a wireless curriculum is natural, given the estimated 400 million phones in

global use today and the growth to 1 billion phones by 2003 [Cellular 2002]. This communication medium has become so pervasive that many developing countries now have a cellular communications infrastructure that are more reliable than their legacy land lines [ITU 2002]. In developed nations, particularly among students and young professionals, cellular phones are increasingly replacing traditional wired phones because they offer mobility, attractive calling rates, and, frequently, digital services such as messaging and Internet facilities [WMPS 2002].

The importance of software comes sharply into focus when these devices are viewed as nodes of a large digital network, with each node having varying degrees of onboard computation power. Java, with its cross-platform capabilities and support for small devices, makes it appealing for applications running on a wide variety of cellular phone hardware and operating systems [Sun 2002b]. Currently, 15 cellular handset manufacturers (Casio, Fujitsu, Hitachi, LG, Panasonic, Mitsubishi, Motorola, NEC, Nokia, RIM, Samsung, Sharp, Siemens, Sony Ericsson and Toshiba [Sun 2002a]) offer 33 models of phones enabled for J2ME, the version of the Java run-time environment that is optimized for small devices. Although this represents a comparatively small percentage of the overall cellular phone market, 80% of phones are expected to support Java by 2006 [Lawton 2002]. J2ME is seen as fueling an “aftermarket” of cellular products, giving people the ability to add capabilities to their phones without having to purchase new ones [Lawton 2002]. Thus, Java is our second building block.

However enabling the synergy between on-phone computers and Java may be, it is not sufficient in itself to ensure high quality, secure, usable products. Wireless device manufacturers are turning to defined software development, user interface, and quality assurance standards in an effort to control the quality of software loaded onto cellular phones at the time of their purchase or downloaded later as a capability upgrade [Nokia 2002e], [Motorola 2002]. It is this aspect – software development practices – that

comprises the third building block of our wireless curriculum. It is this building block that we propose to draw from industry to give our students a rounded exposure to real-world practice.

The problem we will address

Equipping students with knowledge about developing software for wireless devices is problematic in today's academic arena. Colleges teach students the principles of software development, but that instruction is oriented mostly to conventional hardware platforms. Developing software for mobile devices requires specific knowledge in working with hardware that is limited in speed, memory, power, and user interfaces. The traditional university curricula do not address these areas in any depth [Kiely 2001].

We have taken a first step in departing from the traditional curricula by orienting an undergraduate degree to wireless devices. Course material on J2ME for wireless computers, including cellular phones, is readily available. What is missing is the mechanism to expose students to real-world software issues encountered in the wireless software industry. Without this exposure, we and our students can only guess as what makes a piece of software "suitable," especially since wireless software development is such a recent innovation.

What wireless industry practices are available?

Motorola [Motorola 2002], Nokia [Nokia 2002i], and Ericsson [Sony Ericsson 2002] have established processes whereby J2ME third-party software is certified for use on their products. Their certification programs are similar in that they define standards that application software must meet; application programmer interface (API) libraries and tools approved for production use; and procedures by which the application software is tested. Once certified, software may bear a signifying logo as well as be

listed on a registry of products sanctioned as “approved” by the respective wireless manufacturer.

The relevance of this to the academic world is that each of these companies has a publicly available set of guidelines for defining application software suitability. The companies benefit by being able to vet software for usability, robustness, functionality, performance, security, and installation. We feel that universities can benefit from this effort by using the standards as windows into what the industry considers to be “good” software.

All three companies announced their certification programs within six months of each other in 2002. Of the three, Ericsson’s appears to be the least mature. It has the smallest web presence, and details of application standards and testing processes are sketchy. Motorola’s *Application Certification* and Nokia’s *OK* programs are roughly equivalent in rigor; however, Nokia has the edge by offering automated testing tools as well as a more detailed testing process.

The Nokia *OK* process has a number of features that can be cultivated for classroom adaptation. First, Nokia provides a) user interface style guides [Nokia 2002a] which describe the look and feel of applications running on various handset models; b) a J2ME application requirements description [Nokia 2002b] which defines restrictions on application information content, performance latency, operation sequencing, usability, security, and installation; and c) standards for cellular phone games [Nokia 2002c] describing requirements on content, menu operations, scoring, sound, robustness, usability, and installation. Second, it provides APIs, emulators, and testing gremlins for handsets supporting J2ME [Nokia 2002d], [Nokia 2002h]. Third, it provides a test process framework for certifying application suitability [Nokia 2002g].

Since the announcement of the Nokia OK program in mid-2002, 130 J2ME applications have been submitted for certification. Of those, 50 have been certified.

Inability to satisfy usability criteria is the most common reason for not being certified [Nokia 2002e]. Nokia points to 93% of submitted applications failing certification tests, with many requiring up to three re-tests, as an indicator of the success of the program's ability to effectively scrutinize software [Helin 2002]. In forming its application certification program, it is taking advantage of the success from other sectors, such as the Information Technology community in which certification tests have shown a 10-fold increase in reliability over time [Wakid et al 1999], and the software safety community in which certification tests are mandatory and must be successfully passed before software can even be used in a production environment. While cell phone software does not deal with life-critical issues, many of J2ME cell phone applications originate from individuals and small businesses that may not have software development processes which assure products of consistent quality. Nokia's certification efforts model industry leaders that have similar software development populations – Palm, Handspring, and BlackBerry [QP 2002] – and which rely on third-party certification to enforce minimum standards of usability.

Postings on Nokia's "Developer's Discussion Board" (http://nkn.forum.nokia.com) and Slashdot (http://slashdot.org) by in-the-trench software developers give a qualitative insight into the program's success. Nokia's content policies that prevent cellular phone game software from depicting violence and adult themes is viewed as positive; all the developers that indicated that they had undergone the certification tests felt that they paid more attention to the quality of their software than had they not been faced with certification.

What are others doing to incorporate wireless industry practices?

Examination of universities with course catalogs posted on the web, conference proceedings, and research digests show a flurry of research on software development for mobile devices and cellular phones at the graduate level, but very little evidence of

wireless software certification integrated into the undergraduate curriculum. The majority of activity in the undergraduate wireless area comes from application development for cellular phones in isolated senior design projects or in limited offerings of special topics courses. All major universities offer instruction in Java; J2ME-specific instruction appears mostly in upper-level undergraduate courses. Although use of cellular phone J2ME APIs is widespread, we found no universities adapting wireless manufacturer standards, guidelines, or certification processes to the undergraduate classroom. Auburn University is the first university in the nation to offer an undergraduate degree in wireless engineering; we feel we are on leading edge of education in this area.

What makes our ideas unique?

The cellular phone is a tightly constrained computing environment. This can be an advantage to educators because it means that software must be small in size and scope. Students can experience a software development effort from inception to final delivery. It also means that standards and guidelines are small and manageable, and can be distilled for classroom use with little loss of industry intent. This is in sharp contrast to software development for conventional hardware in which standards and guidelines are not comprehensive and are difficult to adapt to student use.

What we plan to do

We propose to adapt the Nokia OK certification concept into the software engineering courses required of our Wireless Engineering majors. The tasks we plan to accomplish mirror the objectives stated earlier, namely:

Task 1: Adapt Nokia OK application standards for classroom use as examples of what is expected of industry-quality software.

Task 2: Incorporate Nokia OK software development and emulation tools into a student-friendly software development environment.

Task 3: Implement in-house Nokia OK certification procedures as a basis for assessing student projects and providing feedback.

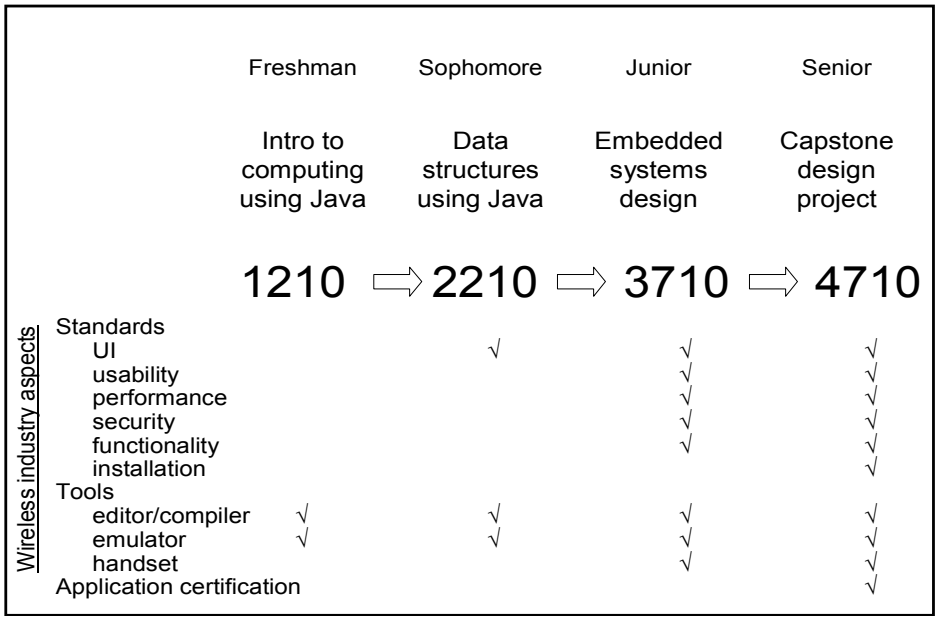

Figure 1. This is the sequence of software engineering courses in the Wireless Engineering software track that will take advantage of the Nokia OK material. The wireless aspects play an introductory role at the end of the first two courses, and a more prominent role in the last two courses.

Our adaptation of Nokia OK will be introduced incrementally across four courses, COMP 1210 (Foundations of Computing I), COMP 2210 (Foundations of Computing II), COMP 3710 (Embedded Systems Development), and COMP 4710 (Senior Design Project), as shown in Figure 1. COMP 1210 and 2210 are introductory software development courses using Java. Through the development and emulation environment of Task 2, we introduce students to the notion that Java is platform independent by illustrating application development with J2ME in COMP 1210. In COMP 2210, students will construct a rudimentary application that adheres to the user interface standards of Task 1. COMP 3710 addresses system-level aspects of wireless

devices. Here students will use Task 1 user interface, performance, and security standards and Task 2 development environment to create applications. They will first test their software using the emulator environment of Task 2, and then test their software with actual hardware. COMP 4710 is the point at which students demonstrate their ability to integrate the knowledge they gained across previous coursework. Here they will develop an application for an actual customer. Their software must adhere to the full complement of standards of Task 1; they will use the development tools of Task 2; and their software will be evaluated internally using the certification process of Task 3.

Task 1: Adapt Nokia OK standards to the classroom

We propose to adapt the documents below into a single lab manual which describes our expectations of the character and quality of student application software.

- Nokia OK MIDP Application Requirements
- Nokia OK MIDP Application Requirements for Games
- Developer Check List for J2ME Applications
- Nokia Series 60 UI Style Guide

References to procedures that are unique to working directly with Nokia will be removed (although, of course, Nokia will be fully acknowledged as the information source). Duplication among the standards will be eliminated, and explanations added to standards are unclear to a non-Nokia audience. The resulting lab manual will articulate standards on user interface look and feel, human-computer interaction guidelines, security requirements, reliability guidelines, performance requirements, content restrictions, installation packaging.

Task 2: Implement Nokia tools into a student-friendly environment

We propose to integrate the Nokia handset Mobile Information Device Profiles (MIDP), J2ME compiler, class libraries, virtual machine (KVM), phone APIs, test

gremlin, and phone emulators into a single integrated development environment, jGRASP.

Although we wish our students to have an exposure to real-world standards, we recognize that we have an obligation to make sure that experience is presented in a pedagogically-sound fashion. We use a particular software development environment, jGRASP, across our computer science, software engineering, and, now, wireless engineering, programs. jGRASP has been demonstrated to be an effective tool for teaching [Cross et al 2001] and learning [Hendrix et al 2002], as indicated by its current user base and its inclusion in Java text books ([Lewis and Loftus 2003] and [Farrell 2003]) . As jGRASP is the intellectual property of Auburn University, we have the ability to tightly integrate other tools into it. Incorporating wireless software libraries, tools, and emulators into jGRASP gives our students the ability to develop cellular phone applications without having to learn another development environment. jGRASP is available at no cost to the software development community at large; the added benefit of this approach is that other educators can take immediate advantage of our work.

We recognize that several J2ME development environments are available. See Table 1 for a comparison. The primary disadvantage to them is that they were not intended for educational use, but rather as professional software development tools, which require a substantial investment in training time. jGRASP, in addition to requiring very little time to learn, has a number of proven pedagogical features built into it, including (1) reverse engineering of UML class diagrams and dependency relationships from source code [Cross and Barowski 2002]; (2) tight integration with a visual debugger for explicating source code [Cross et al 2002b]; and (3) automatic annotation of source code control constructs, control paths, and overall program structure [Cross et al 1998];. The latter has been shown empirically to improve the comprehensibility of source code [Hendrix et al 2002]. In short, jGRASP is a lightweight tool that is rich in

features needed to assist students in engineering software while, at the same time, being unencumbered by unnecessary features required by large-scale professional development. The integration of J2ME into jGRASP will be straightforward, and the synergy of J2ME and the jGRASP student-oriented features will benefit the education community more than using an existing off-the-shelf product.

Table 1: Comparison of jGRASP with other J2ME environments

	jGRASP	JDeveloper	Sun ONE Studio	Nokia Developer's Suite for J2ME
Reference	[GRASP 2002]	[Oracle 2002]	[Sun 2002c]	[Nokia 2002f]
Features	Editing, debugging, compiling	Editing, debugging, compiling, device emulation	Editing, debugging, compiling, device emulation	Editing, debugging, compiling, device emulation
Cost	Free	Free Community Version	Free Community Version	Free
Language support	Ada, C, C++, Java, VHDL	Java (including J2ME)	Java (including J2ME)	J2ME
Resource requirements for tool (not including Java compiler)	3MB RAM [Jain 2002]	56M RAM [Jain 2002]	63MB RAM [Jain 2002]	15 MB RAM
Ease of use	Integrated environment; no tutorial required [Jain 2002]	Integrated environment; no tutorial required [Jain2002]	Integrated environment; tutorial required [Jain 2002]	Loose integration; documentation required.
Special pedagogical features	Control Structure Diagram, collapsible view of code nesting, UML class diagram, visual debugger	UML diagram	None	None

Task 3: Implement a Nokia-like certification process

We propose to take the testing concept of the Nokia OK program and mimic it in the classroom. The actual Nokia OK process has third-party vendors submitting software to Nokia, which, in turn, has an independent laboratory test the software for conformance to standards. We plan to implement this process in the small for capstone design projects built in COMP 4710. Senior design teams will submit their applications

to a graduate students working for the Center for Innovations in Mobile Pervasive Agile Computing Technologies (IMPACT), an Auburn University Peak of Excellence program dedicated to conducting research on ubiquitous computers. Working under the mentorship of a faculty member, this group will follow the Nokia process to test the software and determine its certifiability. No senior design team will be allowed to consider its project complete until it has been certified.

Expected Outcomes and Timetable

We expect the tasks described above to lead to the following outcomes:

- Students will be immersed in a technology similar to what they are likely to encounter in their first years of work.
- Wireless industry standards will be adapted to the classroom, J2ME development capabilities will be incorporated into an existing popular software tool, and an industry testing process will be implemented in an academic setting. These items will be disseminated to the academic community as well as to the wireless industry.
- Evidence of the effectiveness of using industry practices in the classroom, with lessons learned, will be disseminated to the academic community.
- Faculty will become skilled in bridging the gap between the classroom and the wireless software development world.
- A baseline will be created for wireless software engineering education within Auburn University and the university community as a whole.

The timetable for this work is shown in Figure 2.

Facilities and Resources Available

Auburn University provides computing resources to all students, faculty, and staff. Facilities include dial-in access to Unix resources as well as numerous labs equipped with Sun workstations and standard Windows-based desktop computers. Our Wireless Engineering Advisory board has committed to provide the department with cellular phones for COMP 3710 and COMP 4710. We have the in-house resources to fulfill the obligations of this project; we require no additional equipment.

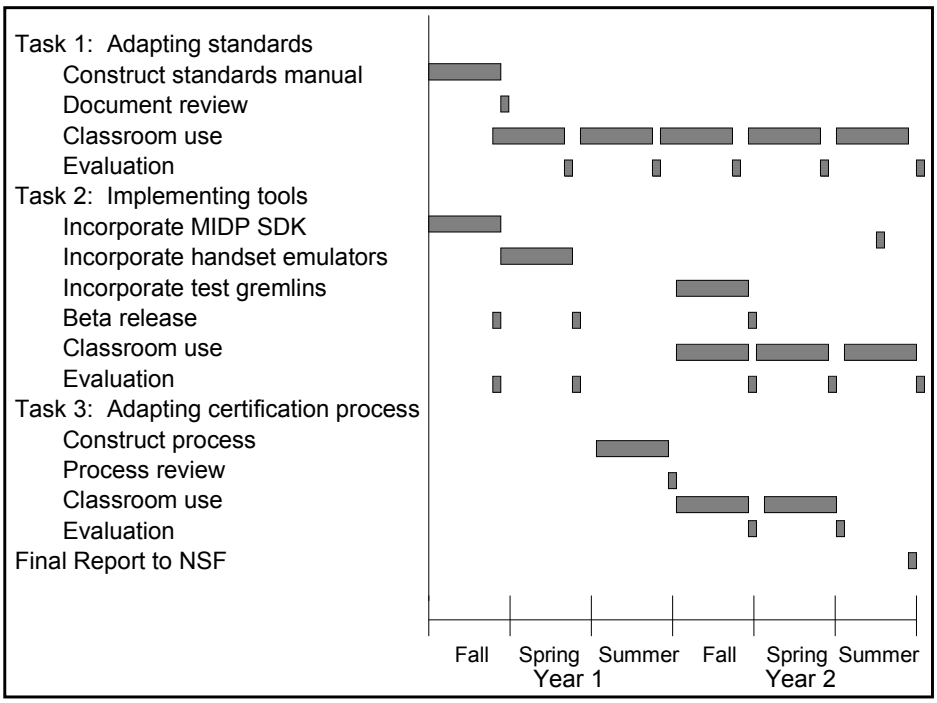

Figure 2. This project will span two years. Standards collection, tool construction, and application certification adaptation will predominate the first year. The second year will conclude tool construction, but will consist primarily of using and evaluating the project products.

Experience and Capability of the PIs

The PIs' academic and industrial qualifications are described in the vita section of this proposal.

Evaluation Plan

First, the department's Undergraduate Curriculum Committee will establish outcomes expected of the software wireless engineering courses, together with quantitative measures for determining the degree to which those outcomes have been met. (Note: This is done for all courses as part of the department's accreditation process.)

Second, all items that result from this project will be reviewed by two groups, one external and one internal, before being released for the first time in the classroom. The external group will be made up of industry members of our Wireless Advisory Board (which includes Vodafone, Hewlitt-Packard, Ericsson, Verizon, Nortel, Texas Instruments, Nokia, and Cingular); the internal group will be 4-to-5 students and faculty drawn from IMPACT. Each group will review each item for accuracy and fidelity to Nokia OK program. No item will be released for use in the classroom until all questions are resolved by both of these groups.

Special attention will be paid to the J2ME-enabled version of jGRASP. The external and internal groups will each receive a beta version for testing. Additionally, since jGRASP will be posted to a web site, downloads will be tracked to gauge the breadth of interest in the prototype. Currently, we are averaging approximately 1000 downloads per week. As a result, we receive a steady stream of comments and suggestions from users. Many of these comments make direct references jGRASP features and will serve as a pool of suggestions for improvement.

Third, once an item has been deemed suitable, it will be used in the courses described earlier in this proposal. A course post mortem will be conducted at the end of each semester for each course that uses the item. Students will complete a Likert scale-type survey indicating their opinion of the relevance, usefulness, usability, and perceived value of the item. Additionally, the faculty involved in the wireless software

engineering courses will provide a self-assessment of the effectiveness of the item in the courses. Both the self assessment and the course post mortem will be distributed to the internal and external review teams for evaluation. The PIs, in conjunction with the course instructors, will address any recommendations for changes and report back to the review teams.

Finally, at the conclusion of the project, the department's Undergraduate Curriculum Committee will conduct an audit of the project by comparing actual student performance to expected course outcomes.

Findings across the entire evaluation process will be included in final project report to NSF.

Dissemination of Results

We wish to share our efforts with whoever is interested in them. To that end, we will disseminate our results as follows:

- **Establish a WWW site.** The site will be updated at each of the project milestones with our most current findings.
- **Materials**. All materials will be available online, including the J2ME-enabled version of jGRASP.
- **Final report.** We will consolidate a final report documenting all project events and make the report available via our web site.
- **Publications**. The investigators plan to present technical papers and demonstrate the software at appropriate computer science and software engineering education conferences, including ACM's SIGCSE Symposium, Frontiers in Education Conference, and the National Educational Computing Conference.

References

[Cellular 2002]	Cellular Phone News. Untitled. http://www.cellularphonenews.com/ebook/overview.html. Nov 2002.
[Cross and Barowski 2002]	Cross, J. H. and L. A. Barowski, "Extraction and Use of Class Dependency Information for Java," Proceedings of the 9th Working Conference on Reverse Engineering, Richmond, VA, October 28 - November 1, 2002, 309-315.
[Cross et al 1998]	Cross, J. H., S. Maghsoodloo, and T. D. Hendrix. "Control Structure Diagrams: Overview and Evaluation," Journal of Empirical Software Engineering, Vol. 3, No. 2, 1998, 131-158.
[Cross et al 1999]	Cross, J. H., T. D. Hendrix, K. S. Mathias, and L. A. Barowski. "Software Visualization and Measurement in Software Engineering Education," Proceedings of Frontiers in Education 1999, November 10-13, 1999, pp. 12b1-5 to 12b1-9.
[Cross et al 2001]	Cross, J. H., T. Dean Hendrix, and Larry A. Barowski, "Debugging [in] CS1," Proceedings of 2001 ACM Southeast Conference, Athens, GA, March 16-17, 2001.
[Cross et al 2002a]	Cross, J. C., T. D. Hendrix, and K. H. Chang. "Scaleable Visualizations to Improve and Measure Comprehensibility of Software Systems: A Framework for Evaluation," NSF award 9806777, June 1998 - June 2002, $845,000.
[Cross et al 2002b]	Cross, J. H., T. D. Hendrix, and L. A. Barowski. "Using the Debugger as an Integral Part of Teaching CS1," Proceedings of Frontiers in Education 2002, November 5-9, 2001, F1G-1 - F1G-6.
[Cross et al 2002c]	Cross, J. H., T. Dean Hendrix, and Larry A. Barowski, "Using UML Class Dependency Diagrams in CS1," Proceedings of 2002 ACM Southeast Conference, Raleigh, NC, April 26-27, 2002, 119-123.
[Farrell 2003]	Farrell, J. *Java Programming*, 2nd Edition. Course Technology.
[GRASP 2002]	GRASP. Auburn University. http://www.eng.auburn.edu/department/csse/research/research_groups/grasp. Nov. 2002.
[Helin 2002]	Helin, R.Nokia OK Program Manager. E-mail correspondence. 4 December 2002.

[Hendrix et al 1998a]	Hendrix, T. D., J. H. Cross, L. A. Barowski, and K. S. Mathias, "Grasping Software," Journal of Computing in Small Colleges, Vol. 13, No. 3, January, 1998, 108-118.
[Hendrix et al 1998b]	Hendrix, T. D., J. H. Cross, L. A. Barowski, and K. S. Mathias, "Visual Support for Incremental Abstraction and Refinement in Ada 95," Proceedings of SIGAda'98, Washington, D.C., November 10-12, 1998.
[Hendrix et al 1998c]	Hendrix, T. D., J. H. Cross, L. A. Barowski, and K. S. Mathias, "Scaleable Visualizations to Support Reverse Engineering," Proceedings of the 5th Working Conference on Reverse Engineering, Honolulu, Hawaii, October 12-14, 1998.
[Hendrix et al 2002]	Hendrix, T. D., J. H. Cross, S. Maghsoodloo, and K. H. Chang. "Empirically Evaluating Scaleable Software Visualizations: An Experimental Framework," IEEE Transactions on Software Engineering, Vol. 28, No. 5, May 2002, 463-477.
[ITU 2002]	International Telecommunication Union. World Telecommunication Development Report. http://www.itu.int/newsarchive/wtdc2002/backgrounder.html. Nov 2002.
[Jain 2002]	Jain, J. Interview regarding observations of run-time memory consumption of J2ME toolkits. Department of Computer Science and Software Engineering. Based on her work for MNGT 7830 – Advanced Database Management Systems. Nov. 2002.
[Kiely 2001]	Kiely, D. 2001. Wanted: Programmers for Handheld Devices. IEEE Computer 34, 5 (May 2001), 12-14
[Lawton 2002]	Lawton, G. Moving Java into mobile phones. *IEEE Computer*, Vol. 35, Number 6, June 2002, pp. 17-20.
[Lewis and Loftus 2003]	Lewis, J., and W. Loftus. *Java Software Solutions*, 3rd Edition. Addison-Wesley.
[Mathias et al 1999]	Mathias, K. S., J. H. Cross, T. D. Hendrix, and L. A. Barowski, "The Role of Software Measures and Metrics in Studies of Program Comprehension," Proceedings of 1999 Southeast ACM Conference.
[Motorola 2002]	Motorola. "Motorola Application Certification Program". Version R1.3A. Available from http://qpqa.com. Nov 2002.
[Nokia 2002a]	Nokia. The Nokia Series 60 Games Style Guide. Accessible from http://forum.nokia.com/main. Nov. 2002

[Nokia 2002b]	Nokia. Nokia OK MIDP Application Requirements v1.1. Accessible from http://forum.nokia.com/main. Nov. 2002.
[Nokia 2002c]	Nokia. Nokia OK MIDP Application Requirements for Games v1.2. Accessible from http://forum.nokia.com/main. Nov. 2002
[Nokia 2002d]	Nokia. Nokia Software Development Tools. http://forum.nokia.com/main. Nov. 2002.
[Nokia 2002e]	Nokia. How to Get the 'Nokia OK' on Your Mobile Application. Nokia Developer Channel. http://portals.devx.com/Nokia/Article/6729. Nov. 2002.
[Nokia 2002f]	Nokia. Nokia Developer's Suite 1.0 for J2ME. http://forum.nokia.com/main/1,35452,030,00.html?fsrParam=2-3-/main/1,35452,030,00.html&fileID=2306. Nov. 2002.
[Nokia 2002g]	Nokia. Nokia OK Terminal Software Application Process Flowchart. http://forum.nokia.com/main. Nov. 2002.
[Nokia 2002h]	Nokia. Nokia Testing Suite. Accessible from http://forum.nokia.com/main. Nov. 2002
[Nokia 2002i]	Nokia. Nokia OK Concept. http://forum.nokia.com/main. Nov 2002.
[Oracle 2002]	Oracle9*i* Jdeveloper 9.0.3. http://technet.oracle.com/products/jdev/htdocs/jdev903_fo.html. Nov. 2002.
[QP 2002]	Quality Partners. Custom Compliance Programs. http://qpqa.com/qpspecial.htm.
[Sony Ericsson 2002]	Sony Ericsson. Sony Ericsson to deliver tools and services to wireless developers with Metrowerks. Press Release. http://www.sonyericsson.com/spg.jsp?page=gis&Redir=page%3DC2_1_26%26B%3Dns. 18 March 2002.
[Sun 2002a]	Sun Microsystems. " First Understand Your Market." http://wireless.java.sun.com/deploy/market.html. Nov 2002.
[Sun 2002b]	Sun Microsystems. JavaTM 2 Platform, Micro Edition (J2METM)http://java.sun.com/j2me. Nov 2002.
[Sun 2002c]	Sun™ ONE Studio 4 update 1. http://wwws.sun.com/software/sundev/jde/index.html. Nov. 2002.
[Wakid et al	Wakid, S., D. Kuhn, and D. Wallace. Toward Credible IT Testing and

1999] Certification. *IEEE Software, 16,* 4, pp. 39-47. July/August 1999.

[WEREC 2001] WEREC. Wireless Engineering Research And Education Center (WEREC) Whitepaper. 2001. College of Engineering, Auburn University.

[WMPS 2002] Wireless Mobile Phone Search. "Cell phone vs land lines." http://www.wireless-mobile-phone-search.com/cell_vs_wireless.htm. Nov. 2002

Project Participant David A. Umphress, Ph.D.
Computer Science and Software Engineering
107 Dunstan Hall
Auburn University All 36849
Voice: (334) 844-6335
Fax: (334) 844-6329
e-mail: umphress@eng.auburn.edu
WWW: http://www.eng.auburn.edu/~umphress

Professional Preparation

Angelo State University	Computer Science	B.S. 1977
Texas A&M University	Computer Science	M.C.S 1978
Texas A&M University	Computer Science	Ph.D. 1987

Appointments

1999-	Associate Professor, Auburn University
1997-99	Associate Professor and Department Chair, Seattle University
1993-97	Associate Professor and Director of Software Engineering, Seattle University
1991-93	Lead Software Engineer, US Strategic Command, USAF
1987-91	Assistant Professor, Air Force Institute of Technology, USAF
1983-87	Communications-Computer Officer, Texas A&M University, USAF
1978-83	Lead Systems Programmer, 1020 Computer Services Squadron, USAF

Publications

Related to Proposal

Umphress, D., T. Hendrix, and J. Cross. 2002. Software process in the classroom: the capstone project experience. *IEEE Software, 19*, 5, 78-85.

Umphress, D. and J. Hamilton, Jr. 2002. Software process as a foundation for teaching, learning, and accrediting. *Proceedings of the 2002 Conference on Software Engineering Education and Training*. pp. 160-169.

Umphress, D., and R. Chapman. 2001. Selecting a PDA for teaching software development. CSSE Technical Report. Auburn University AL.

Umphress, D. 1997. Experiences in constructing a level-2 software engineering graduate curriculum. Proceedings of the 10th Conference on Software Engineering Education (Virginia Beach, VA, April). IEEE Press. 4-12.

Umphress, D., V. Helbling, J. Russell, and C. Keene. 1995. Software process maturation. Information Systems Management 12, 2 (Spring 95), 32-42

Other Publications

Loughry, J., and D. Umphress. 2002. Information Leakage from Optical Emanations. *ACM Transactions on Information and System Security*, 5, 3, 262-289.

Kirner, D., R. Porter, P. Punniamoorthy, M. Schuh, D. Shoup, S. Tindall, D. Umphress. 1999. Extending Use Cases Throughout the Software Lifecycle. *Software Engineering Notes 24*, 3, 66-68.

Dingle, A., Umphress, D., and R. Moul. 1998. Explication of Object-Oriented Databases: A Software Engineering Perspective. *Proceedings of the 29th SIGCSE Technical Symposium on Computer Science Education* (Atlanta, GA, February). ACM Press. 35-39.

Umphress, D., and S. March. 1991. Object-oriented requirements determination. *Journal of Object-Oriented Programming*. Winter Special Edition. pp. 35-40.

Spicer, K., and D. Umphress. 1991. A method for mapping an analysis to a reusable design. *Ada Letters XI*, 9, 67-82.

Synergistic Activities

- Developed and taught software development course for handheld computers. 2002.
- Injected small-team processes (e.g., Team Support Process, eXtreme Programming) into software engineering undergraduate and graduate projects. Developed and taught a software process course designed to expose students to small- and large-team processes. 1993-present
- Worked as a process consultant advising software development organizations in preparation for CMM and ISO process assessments. 1990-1996
- Led joint Air Force-Navy venture to modernize 900+ person software organization. Efforts evidenced by organizational reassessment from CMM Level 1 to Level 2,with strong inroads to Level 3. 1992-6
- Led software development of 150KLOC workstation application in Ada. By-product of the effort was proof-of-concept of team software processes in a military culture. 1991-92

Collaborators & Other Affiliations

Collaborators

Carl Chang	AuburnUniversity and University of Illinois at Chicago
Kai Chang	Auburn University
Richard Chapman	Auburn University
James H. Cross II	Auburn University
Adair Dingle	Seattle University
John A. Hamilton Jr	Auburn University
T. Dean Hendrix	Auburn University
Jane Huang	University of Illinois at Chicago
Steve McConnell	Construx Software

Graduate and Postdoctoral Advisors

Doctoral Co-Advisors: Dr Udo Pooch and Dr Glen Williams Texas A&M Univ
PostDoc: NA

Thesis Advisor and Postgraduate-Scholar Sponsor

Ross Porter	Raincity Technology
Joe Loughry	Lockheed-Martin

Biographical Sketch

Principal Investigator James H. Cross II, Ph.D., Professor and Chair

Computer Science and Software Engineering
107 Dunstan Hall
Auburn University, AL 36849
(334) 844-6315 cross@eng.auburn.edu

Education
B.S. Mathematics, University of Houston, 1971
M.S. Mathematics, Sam Houston State University, 1976
Ph.D. Computer Science, Texas A&M University, 1986

Experience
1997- Professor, Auburn University
1996- CSSE Dept. Chair, Auburn University
1991-96 Associate Professor, Auburn University
1986-91 Assistant Professor, Auburn University
1985-86 Research Associate and Lecturer, Texas A&M University
1981-84 Chief, Computer Services, DoD Dependents Schools-Atlantic Region
1979-81 Computer Coordinator, DoD Dependents Schools-Pacific Region
1977-79 Computer Specialist, DoD Dependents Schools-Pacific Region
1976-77 Computer Science Teacher, DoD Dependents Schools-Pacific Region
1973-76 Technical Training Instructor, Houston Technical Institute
1969-73 Computer Operator/Programmer, Texaco, Inc., Bellaire, Texas

Areas of Interest

Dr. Cross is primarily interested in teaching undergraduate and graduate courses in software engineering and directing research in the areas of software methodology, testing and reverse engineering. His continuing research efforts include the GRASP project, which focuses on reverse engineering and the automatic generation of graphical representations of software. The purpose of the generated visualizations is to improve the comprehensibility of software, and as a result, increase productivity and reliability across all activities that involve code reading.

Publications Related to the Proposal

1) J. H. Cross, T. D. Hendrix, and L. A. Barowski. Using the Debugger as an Integral Part of Teaching CS 1. *Proceedings of Frontiers in Education 2002*, November 6-9, 2002, Boston, MA.

2) T. D. Hendrix, J. H. Cross, and S. Maghsoodloo, The Effectiveness of Control Structure Diagrams in Source Code Comprehension Activities, *IEEE Transactions on Software Engineering*, Vol. 28, No. 5, pp. 1-15, May 2002.

3) T. D. Hendrix, J. H. Cross, and S. Maghsoodloo. An Experimental Validation of Control Structure Diagrams. *Proceedings of the 7th Working Conference on Reverse Engineering*, Brisbane, Australia, November 23 – 25, 2000.

4) J. H. Cross, S. Maghsoodloo, and T. D. Hendrix. The Control Structure Diagram: An Overview and Initial Evaluation. *Empirical Software Engineering*, Vol. 3, No. 2, pp. 131-158, 1998.

5) J. H. Cross, K. H. Chang, T. D. Hendrix, R.O. Chapman, and P. A. McQuaid, "Visualization and Measurement of Source Code," *CrossTalk Journal of Defense Software Engineering*, December 1997, 16-19.

6) J. H. Cross and R. S. Dannelly, "Reverse Engineering Graphical Representations of X Source Code," *International Journal of Software Engineering and Knowledge Engineering*, Vol. 6, No. 2, June 1996, 1-21.

7) J. H. Cross, E. J. Chikofsky and C. H. May, "Reverse Engineering," *Advances in Computers*, Vol. 35, 1992, 199-254.

FACILITIES, EQUIPMENT & OTHER RESOURCES

FACILITIES: Identify the facilities to be used at each performance site listed and, as appropriate, indicate their capacities, pertinent capabilities, relative proximity, and extent of availability to the project. Use "Other" to describe the facilities at any other performance sites listed and at sites for field studies. USE additional pages as necessary.

Laboratory: **Laboratory space for this project is provided by the Department of Computer Science and Software Engineering.**

Clinical:

Animal:

Computer: **The Samuel Ginn College of Engineering provides 300+ Sun workstations, 26 Sun servers with a terabyte of online disk storage and two quad processor comptuer servers. The College also supports 800+ fully integrated PCs. All faculty have computer equipment with network access in their office.**

Office:

Other: _____

MAJOR EQUIPMENT: List the most important items available for this project and, as appropriate identifying the location and pertinent capabilities of each.

OTHER RESOURCES: Provide any information describing the other resources available for the project. Identify support services such as consultant, secretarial, machine shop, and electronics shop, and the extent to which they will be available for the project. Include an explanation of any consortium/contractual arrangements with other organizations.

NOKIA

National Science Foundation
4201 Wilson Blvd
Arlington VA 22230

November 26, 2002

Re: Auburn University's Proposal to Program Solicitation NSF-02-095

Dear NSF Reviewers:

I enthusiastically endorse Auburn University's proposal to adapt a wireless software certification process for classroom use. Auburn stepped out this year as the first university in the country to offer a degree that prepares students for wireless careers. I believe the adaptation of wireless software certification processes is another bold move, one that will enable students to get a jumpstart on what will be ahead of them once they enter the workforce.

Cellular phone callers are not just using a piece of hardware when they pick up a Nokia handset. They are using a computer that enables them to talk with other people, play games, receive text messages, and read their e-mail. It is software that makes this possible. As we move into the future, software will increasingly become the enabling technology for advances in cellular phone technology.

At Nokia, we insist that our hardware pass rigid certification criteria before being released to the consumer. We follow similar qualification procedures for the software we develop internally. Our ability to contain software quality within Nokia has been relatively easy in the past since application development required skills and tools available to a small audience. With the exploding growth of Java-enabled phones, we recognize that software applications will come more and more from third-party vendors which may not necessarily adhere to Nokia quality practices. Application certification is an effective filter for ensuring that the software that runs on Nokia phones has a consistent user presentation, is reliable, and is secure.

Implementing a certification process in the classroom will have a similar effect: it will let students know in advance what standards they are expected to uphold, then will ensure they meet those standards. This has two side effects. First, students will enter wireless careers understanding a culture of quality. Second, the industry will get feedback from the project on the academic usefulness and usability of the certification process itself.

Nokia, Inc.
6000 Connection Drive
Irving, Texas 75039

Tel. 972 894 5000
Fax 972 894 5050

I urge you to consider seriously Auburn University's proposal.

Yours sincerely,

Donal O'Connell
Vice President, NMP R & D Texas

NOKIA
6000 Connection Drive
Irving, Texas 75039

Tel. 972 894 5000
Fax 972 894 5050