

Final Report:

Accelerated Mobile Product Development –
An Economic Engine for Science and Technology Superiority

Submitted by:

PI: Ravi Shankar, Professor and Director, CSI, FAU
Co-PI: Ankur Agarwal, Assistant Professor, CEECS, FAU
Co-PI: Georgiana Hamza-Lup, Assistant Professor, CEECS, FAU
Faculty Member: Prof. Fran McAfee, MFA, Arts and Letters, FAU
Faculty Member: Prof. Norman Silva, MFA, Arts and Letters, FAU
PhD Student: Mihai Fonage, CEECS, FAU
Undergrad Student: Charles Norona
Undergrad Student: Victor Gallego
Undergrad Student: Alex Andresen

Contact Information:

Ravi Shankar, shankar@fau.edu, (561) 297-3470/ 297-2800

Center for Systems Integration
Department of Computer and Electrical Engineering and Computer Science
Florida Atlantic University
Boca Raton, FL 33431

Submitted on: November 22, 2010

Table of Contents

I. Abstract.....	3
II. Introduction	6
III. Significance.....	11
IV. Specific Objectives	12
V. Methods and Results	13
VI. Discussion.....	14
VII. Appendices	16
Appendix A: Software-Hardware Codesign Documents	16
Appendix B: Android Projects Documents.....	16
B1. Android Arduino Bluetooth Quick Communication Setup.....	16
B2. Android Installation and Use.....	18
B3. ATMEGA References – 9/16/10.....	18
B4. More References	18
B4. References to the Robotics Course.....	20
B5. References – 9/8/10.....	21
B6. Websites List	21
References.....	22

I. Abstract

Our stated goal has been to develop an incubation center that is amenable to the background and interests of high school students, and that allows them to rapidly and easily develop high technology products/services that have potential for commercialization. This type of incubation of young entrepreneurial minds will spawn new information intensive products and companies in the longer run – much similar to Steve Job’s Apple, Bill Gate’s Microsoft, Yahoo, Google, Face Book, etc., in a more structured and nurtured manner. Further, it is also our belief that an ideal entrepreneurial team consists of several people with complementary background and interest in creativity/arts, science and technology, and marketing and management. Thus, we included three students in each team, each with at least one of the above cited interests. To ensure success of such teams, we needed to facilitate the learning and interaction of these students with these diverse perspectives and interests. We chose Google’s Android Mobile Platform with this very goal in mind. The platform provides separation of these perspectives, so one can focus on each of them individually, while as a team, they build a software application or a system that has state-of-the-art technology support (in speed, memory, portability, internet access, wireless connectivity, standards, various displays, etc.,).

Prior to start of this funding, during summer ’09, we developed material for a course on developing applications on the Android platform. We have also developed many innovative ideas with the intent to automate and simplify the development process. We offered this course to our engineering undergraduate students in fall ’09 semester. The intent was to learn from this course offering, and improve the course presentation, material, and the tools used, so we will be better prepared for teaching the material to high school students during summer ’10. Typically, a spring semester is a busy period for high school students. Thus, we decided to offer the design and development course during summer ’10 to high school students. During the spring ’10 semester, we offered two courses, one each focused on application development and component /framework development. The former is a continuation of the fall ’09 course, with substantial changes: the focus now was on developing more sophisticated gaming and social networking applications, and with better user interface, thus integrating functionality and creativity. The undergraduate students worked in groups of two. Guest lectures by a professor with expertise in visual communication were provided. He presented techniques and tools to improve the UI (user interface) and demonstrated the use of several graphics tools. The latter course had both a graduate and an undergraduate section, and is focused on building a library of optimized software components/ frameworks / libraries details of which are available at our blog site (<http://android.fau.edu/>). We decided to provide as much of our work as possible at this site (we have documented fully eight applications at the site; they are ready to be downloaded, modified, and/or executed). This is our contribution to the open source community of Android. The undergraduate students in this course typically undertook development of software components, while graduate students worked in groups of three to develop application frameworks. The former class had an enrolment of 15 (due to lab size limitation) undergraduate students, while the latter course had an enrollment of 35 graduate and undergraduate students. Many of the students in the latter group are distance learning students. During the spring semester, the university approved the summer ’10 course for high school students; we were aided by the DESS (Division of Engineering Student Services) on campus to recruit top students from a diverse student population; and we enrolled 30 high school students.

During the summer semester, we actually offered the three-week course. Twenty eight students attended, with the other two dropping out for reasons unrelated to the course. The course was a tremendous success. We had four faculty member, five engineering and one business student

mentors, and one administrator who helped the high school students through every aspect of the course. Each team comprised of three students, one each with creativity/arts, science, and business background/interest. Each team had a student doubling as a project coordinator who was considered to be the spokesperson for the team and who managed the various team responsibilities and tracked the goals. Their stated goal was to develop a marketable mobile application for Android. During the three week program, classes were scheduled MWF from 9.30 AM to 4.30 PM, with an hour for lunch in-between. However, because of the short duration of the scheduled periods, we planned additional meetings and discussion sessions many of the Tuesdays and Thursdays. The students willingly came and worked with us, as they were motivated and excited to participate, and to make sure they had a complete project by the end of the three-week period.

Scholarship awards were made based on ranking of the teams. A group of eight judges scored the teams on the last day (7/2/10), on a scale of 0 to 10, with 10 being the best score. Our (teaching and mentoring) team provided internal assessments worth 30 points, with 10 points set aside for each of the three aspects of the project: Programming, Graphics, and Marketing Video. This internal assessment was based on our constant interaction with the students; we scored them high if they did most of the work (that is, they did not depend upon the professors and mentors too much) and if it was their own work. As indicated above, all these teams were supported by a core group of four faculty members (with engineering and arts backgrounds) and six engineering/business students as mentors. Four of these mentors were assigned specific groups to work with and they provided all the needed technical help. One other engineering mentor coordinated the effort with DESS which offered this course as one of their five courses – and we had an obligation to ensure that their requirements were met as well. The sixth mentor was a business undergraduate student (a volunteer) who advised the students on project management, presentation skills, and helped coordinate the final day event. A total of \$12,000 in scholarships was awarded to nine groups on the final day, based on an aggregate score of judges' scores and our internal assessment. The College of Engineering and Computer Science also presented framed pictures of engineering art as recognition for group excellence in engineering design, and in graphics and animation. We also presented one award of individual excellence.

Subsequent to the conclusion of the course, we evaluated the applications and determined that one of them is immediately marketable; and that six of the remaining eight can be marketed with some extra work. The university is at present supporting four of these mentors during the fall semester so we can complete the work and market at least seven of those applications. Our intent is to accumulate enough funds to offer this course to high school students again next summer, and distribute the remaining funds, if any, to the authors (the high school students, mentors, and the faculty members involved) as per the university's IP policy.

In summary, all efforts are on track for offering again a summer course to high school students and for sustaining the model in the longer run.

The enclosed report documents all our developments. Under 'Introduction' we provide information on our Center for Systems Integration (CSI). CSI has had significant amount of industrial funding, during the period of 2003 to 2008, to radically increase engineering design productivity. We have incorporated many of those ideas in this project. This section also provides more details on the conduct of this project. We believe that we were able to manage this intensive course so effectively because of our earlier work on increasing engineering design productivity. Under 'Significance' we list the benefits of our approach. The section on 'Specific Objectives' lists the schedule and milestones to be achieved during this one year of funding. Under 'Methods and Results' our focus in this report is on activities since the submission of the last quarterly

report. In earlier reports, we covered the following: (1) Development of course modules during summer '09, prior to the start of the proposal; (2) Documentation of the first course offering (in fall '09), in terms of class material and good student designs; (3) Documentation of the second set of course offerings (in spring '10); and (4) Offering of the Android course to high school students this summer and consequent development of 9 marketable applications. We are currently working on the fifth goal of achieving sustenance of the training process via commercialization of the developed Android Applications. Several of our engineering students and high school students have either started or shown inclination to start businesses focused on application development. One student group submitted a SBIR proposal, with our university as a partner. We hope South Florida will become a center for cottage industry in mobile application development.

The section on 'Discussion' was earlier limited to a discussion of our experiences from teaching our engineering courses. We have now expanded this and previous sections in this report with details on innovations and adaptation of the material to better suit the high school students. Once again, this report is limited to progress made since the last quarterly report. Please see the quarterly reports for more information in earlier quarters.

Android is a rapidly evolving development environment. For example, when we started the course development during summer '09, we used version 1.0, which rapidly evolved to version 1.6 by the end of the fall '09 semester, and the current release is version 2.1. With later releases increased sophistication has come in both software and hardware. Now, Android supports multi-touch user interaction and is supported by hardware running at 1 GHz and with 256 Mb or more of memory. This provides both an enormous opportunity (in developing sophisticated applications) and a daunting challenge (since these applications have to be optimized for mobile phones. Response time, battery power, screen size, intermittent network connection, etc., are some of the major issues). During the second quarter, we also purchased newer Nexus One phones, with both SBA grant and department funds.

We had provided detailed appendices in our first two quarterly reports. We replaced those, in our third quarter report, with links to an university blog site where all the details were provided. This site in turn linked to all student blog sites, where the students had documented work on their projects. See <http://android.fau.edu/documents/> - In the fourth quarter report, we consolidated all the student reports on one password protected page at our website, android.fau.edu. We have already provided these details. We did this consolidation so we can take these applications further and market them.

At present, efforts are underway to market these Android Apps. Four of them will be available within a week, with 3 or 4 more expected to be ready by mid December, 2010. We hope this process will help us sustain the program in the longer run.

During the fall semester, we taught our course on Android applications again, for a group of 26 students. The students improved the applications developed this past summer by the high school students and also learnt the use of NDK (native development kit) so they could combine C and Java code, the combination of which has high potential to lead to optimized performance. We expect to use both Java and C in our next semester course offering; we will focus on games with social impact. We also started teaching a course on physical computing in which Android phones are used to remotely manage a set of robots. For more information, see: robotics.fau.edu. We plan to expand this course to a two course sequence, starting in spring '11. This is also expected to lead to a good summer course for middle and high school students during forthcoming summer semesters.

II. Introduction

Our college-level Center for Systems Integration (www.csi.fau.edu) at FAU is dedicated to significant acceleration of the development of mobile embedded systems. The silicon and consumer trends lead to ever more complex embedded systems. A new system designed in 2010 may take 10 years to develop. Motorola, Plantation, FL, has funded us during 2003 to 2008 (for a total of \$1.05 M) to develop a methodology (entitled 'One Pass to Production' or OPP) that is envisioned to reduce the product development time about 700 fold, from 24 months (the cycle time in 2003) to 24 hours. We have made significant progress, reducing the cycle time to 3 months. Motorola hired 7 of OPP graduates to work in their elite APD (Advanced Product Development) team. These students have since then moved on to other local and non-local communication companies because of changed focus at Motorola and disbandment of the APD team¹.

Accelerated product development will help address the needs of the consumer in a timely manner (imagine predicting the consumer behavior 2 years ahead of time, as is the case today) and gain a larger share of that market by first mover advantage. A 700 fold improvement in productivity might seem unrealistic, but consider the following products that have seen remarkable exponential progress: electronic chip complexity and capacity (a million fold improvement in 30 years, 1974 to 2004); engineering design automation (EDA) involved in designing chips (1000 fold in 30 years, 1974 to 2004); and DNA sequencing (10,000 fold improvement in 35 years, 1965 to 2000).

Our methodology combines a top-down approach to capture the desires and constraints of the users in an abstract manner, with an architect translating it to a state-of-the-art system by combining judiciously software, hardware, mechanical, and sensor components to secure a system, optimized perhaps for high performance, long battery life, small size, seamless integration and networking with other devices, etc. The current industry flow suffers from communication gaps, unnecessary iterations, and incompatible components; this delays product launch and quality. By allowing the stakeholders to pick and choose pre-designed / characterized components, and validating that they work well together, this process becomes less error-prone and rapid. Rapid prototyping of mechanical devices (via 3-D printing), reconfigurable robots, and ad-hoc networking of various communication devices contribute to a feasible environment for developing hand-held, portable, and mobile products quickly and in a dependable manner.

The Potential: The current trends and economic successes (such as at Google and Facebook) show the need to integrate the talents of artists, entrepreneurs, and scientists. Google continues to attract the most talented computer programmers and probably builds a pipeline of eager converts to the field, in the process. We propose to use our OPP infrastructure to help our next generation of scientists, artists, and entrepreneurs to integrate their efforts and passions to build systems that are unique and targeted. By providing an environment of accelerated product development, tuning the product to the application domain is fairly immediate. Use of top-down approaches allows the

¹ Motorola now has a strong focus on developing hardware and applications for Android. We are using MotoDev, Motorola's development environment, with the intent to help our graduating students find jobs with local and non-local Motorola branches, and/or become independent application developers as part of Motorola supply chain. Of course, the students will be able to contribute to other companies' portfolios as well as the Android Market, a commercial website that hosts all Android applications

creative student to develop a product at a high level of abstraction, while the use of bottom-up approaches allows the science oriented student to build components to realize these abstractions. Add an entrepreneurial student to the mix, and the appropriate support of business experts, a product could be marketed quickly. Mass customization becomes easier².

Thus a complete eco system can evolve: For creative individuals, an environment to develop high-tech mobile products with the least apprehension for the technology; For the science and technology oriented student, an environment to provide high-tech products (components, platforms, applications, algorithms, and libraries) and services (software and hardware tools for engineering design automation); and for the entrepreneurial student, an avenue to incorporate high-tech in their enterprise and make the product relevant to today's market. Of course, right in the middle of all of this, a new breed of engineers and architects will evolve who will specialize in systems issues and who can interface with the other three categories. These systems specialists, so called systems integrators by IBM in a different context, will bridge the gap between the worlds of creative, scientific, and business oriented students and individuals³.

Rationale behind the Choice: We chose Google's Android Mobile Platform for developing mobile applications because of several reasons: it is perhaps the most exciting platform that is also open source; they provide APIs (application programming interfaces) that make it easy to incorporate complex functionality of the Android easily into one's application software; a large number of applications of interest to high school students, such as games, social web, location based services, etc., can be built; applications developed have the most potential to succeed since most leading mobile phone developers and service providers (such as Motorola, Samsung, and T-Mobile) are using the Android platform; and high school students probably can unleash their creativity better with such a multi-media product that they always carry. Also, unlike iPhone, the Android platform was introduced only recently, giving an opportunity for new comers to contribute to and to benefit from. Android is also benefitting from the Open Set Alliance, with published hardware platform architecture, so hardware manufacturers can unleash their creativity. Combination of Microsoft APIs and the open PC architecture drove the PC industry to new heights. We expect the same kind of innovation to come to the Mobile phone world. While Apple continues to innovate, their proprietary stance discourages everyone's participation. Further, we have made much of our code open source so we can contribute to this atmosphere of innovation and further refinements, which will help the society in general. It will also enhance our 'brand name' so our students will find better jobs and more opportunities. Finally, Android OS (operating system) is not uniquely limited to mobile systems. It is expected to appear on home electronics and appliances, and even PCs. So, the knowledge gained will help the students to migrate to other application domains fairly easily.

Challenges with the Choice: However, there are equally daunting reasons why the Android Platform is not suitable for high school students: It uses Java for programming, which very few

² Android has a rich collection of APIs (application programming interface) that give access to well written software components and well integrated hardware components. Thus, a full fledged bottom-up approach is replaced with a component-based design process that is rapid. By providing a rich collection, Android avoided making the APIs too generic which would have made them too top-heavy and not suitable for mobile applications. However, there is a need to quickly search through the long API suite to pick the right ones.

³ Android separates UI (user interface) development from the development of the functional logic, thus allowing the creative individuals and the technology savvy individuals to simultaneously make progress and iteratively develop the final system. The business oriented individuals will be able to help identify the market niche and develop promotional videos. This process has been started in our spring' 10 course offering.

high school students learn by the time they graduate; The APIs of Android number into thousands, requiring a long learning curve (but can be learnt on an incremental basis); Android also uses XML technology, a step beyond HTML used in web publishing; the resulting code can be long and confusing, especially with the computer scientists' habit of using similar labels for different types of building blocks. Most importantly, the conventional way of designing an application continues to be bottom-up, that is trial-and-error based,; and the development process itself can be frustrating for someone who is not familiar with the Java language and the APIs. As is typical in the industry, this process invariably leads to a compromise in features and to a product that is less-than-desirable from the user perspective⁴.

Approach Taken: These are major incompatibilities in the short run. High school students might lose motivation and interest if the application development takes a long time and/or is compromised in some major way. One needs to take a fresh approach. We have used our experiences from an industry funded project to develop this fresh approach. Over the past 6 years, we were funded by a major cell phone developer to develop a process to rapidly design and develop new cell phones. The process is called OPP – One Pass to Production; see www.csi.fau.edu for our publications. The company has adopted several of our recommendations. If our flow is fully implemented, they can potentially reduce the product development time by 75% or more, and achieve this with less experienced engineers. All the building blocks for our approach exist today, but are spread out in different engineering and business domains, requiring a major overhaul in how products are developed today to acquire and integrate those expertise and skills. As an aside, such drastic improvements may appear unreachable. However, consider this: Intel reduced power dissipation on their chips 7000 fold over the past 35 years. EDA (engineering design automation) tools used in chip design is another example where the productivity increased 1000 fold over 30 years. All that is required is a focus and a commitment to reach that goal.

We decided to focus and commit ourselves to the cause of reaching out to high school students and enabling them to be more productive in a highly challenging high tech field. We have shown that our process can indeed make this possible. It consisted of both a top-down approach of user-centric design and a bottom-up approach of component-based design⁵, coupled with a middle-out approach to optimize the design, based on a certain set of QOS (quality of service) metrics (this last part was not fully utilized in this summer course to high school students). A rapidly prototyped application enables the developers to demonstrate sooner the concept to businesses and raise capital to improve the product and /or commercialize it. Our approach has the following elements: Top-Down Design with application and technical story boarding, to capture user intentions faithfully and to design robust and reliable systems (we have taught a course based on this concept and have had several good designs to share. See www.csi.fau.edu); Automation

⁴Android, on the other hand, encourages not only a top-down design process, with a well-supported bottom-up component level design, but also a user-perspective that is missing in typical computer programming courses. Over the past 9 months, we have already witnessed many innovative ideas on helping newcomers to the Android world. This will only get better. So, we expect the challenges to diminish in importance quite rapidly. However, there is potential for many disappointed application developers, if they do not pay attention to developing optimized code. A sophisticated application that may execute well on a desktop PC would drain the battery and /or freeze on a mobile phone. Code optimization was a major focus of our second course in spring '10.

⁵ Android provides support for many components via well designed and non-generic APIs. There is substantial potential for application optimization, as a consequence. However, the use of Java for business or functional logic code development and the use of XML files for all layout related aspects, need to be carefully monitored and managed. Much work lies ahead for development of design principles for Android code development.

Tools to help the designer make rapid selections (our engineering students developed this semester various frameworks and libraries to facilitate this); Integration support from skilled engineers and programmers (in our case, it was provided by undergraduate and graduate engineering student mentors); and incorporation of aesthetics in the application and promotional video production so the application software is ready for commercialization (this was provided by two professors from the college of arts and letters). All these goals are listed in the original proposal submitted, on page 4.

Incubation Center: We provide here the steps that we will implement / have implemented in reaching our overarching goal: During **summer '09**, prior to this funding, a team of professors and students worked diligently to develop a course on the Android Platform, using UML for top-down design, Java for programming, and Eclipse for application integration and debugging. The course has developed an illustrative application example based on this approach and has developed assignments for another application example, so students can develop this second example in a step-by-step manner. This course was offered in **fall '09** to our undergraduate and graduate students. We monitored their progress to determine how best to change the course and identify barriers to rapid learning. Our **spring '10** semester courses benefitted from this, as already detailed above. The course on application design ('Software-Hardware CoDesign') was based on two-student teams to develop more sophisticated applications that are also aesthetically pleasing. Professor McAfee, from the College of Arts and Letters, gave guest lectures on appropriate visual communication techniques. The course on component design ('Android Software Component Development') focused on developing frameworks⁶ for applications and on optimizing components⁷.

Our **spring '10 work** yielded several applications that have commercial potential. These were improved further by high school students this summer, as detailed below.

We offered our course to high school students during **summer '10**. We used top-down design, automation tools, tutorials, software component library, UI design, and existing applications as the starting point; we also used/introduced software tools for graphics, animation, and video shooting and edition. All these helped the high school students develop their applications rapidly, in a commercially relevant manner. Generally speaking, a spring semester is a demanding semester for high school students (SAT and AP Exams are scheduled then), and thus we planned to offer the course to high school students during summer '10 only. We recruited 30 high school students from various local high schools, of which 28 attended the course. These students brought creative, technical, and business perspectives to form teams; this facilitated formation of typical business environments/cultures. In order to encourage students to excel, we established a panel of judges to evaluate the students' work at the end of the three-week course. We used these rankings to award students scholarships of \$100 to \$800 each. For more details, see: <http://www.dess.eng.fau.edu/pdf/esp/EGN1935android.pdf> Note: In the state of Florida, high school students enrolling for college credit are paid their tuition fees, if the student has a GPA of 3.0 or better.

During **fall '10** the university funded four engineering students (who were mentors for the high school students during summer '10) to improve the applications. The four professors involved guided the students, and critically reviewed the work and the Apps. As of writing this report, four

⁶ One can consider these as 'super components'. Availability of such components will help novice users to feel comfortable in building sophisticated applications.

⁷ This will help meet user QOS (quality of service) metrics, such as latency, response time, battery power, etc.

of the Apps are ready to be marketed, with 3 or 4 more expected to be ready by December 15, 2010. The university will use the Android on-line market to reach customers. The revenue raised will be used to sustain the program at the university, as described above and below

Future Plans: Our focus here is on building on the momentum toward a model of sustenance. Here are some of our plans/actions in that regard: (1) We expect to market at least 7 of the 9 applications developed by the high school students during summer '10. They have needed some extra work, of about 100 hours each by one of the engineering student mentors involved in the application development. The Office of Technology Transfer at FAU has completed most of the work to market these Apps. (2) We are offering a course on Android Robotics (the course is entitled 'Android Projects') during the fall'10 semester. This has allowed us to attract electrical engineering students to the fold. Our goal is to build microcontroller-powered autonomous robots and to coordinate their activities with various low cost and low power communication mechanisms, both among the robots and with Android phones. (3) We expect this to provide the basis for a summer'11 course to be offered to middle school students, with the goal to teach them principles of Math and Physics. A proposal is due to be submitted to NSF in January 2011. (4) We expect to submit a proposal to NCIIA to develop innovative curriculum centered on Android mobile phones. We expect to focus on games with social impact, which will require synergistic use of our various expertise in graphics, animation, and code development at both Java and C. In addition, a professor from Anthropology will participate and advise us in ensuring that the applications will be relevant and appropriate. Our goal is to enhance the interest of high school students in science and engineering, and prepare them to develop sophisticated and marketable applications, and start small businesses that make them money. (5) We identified from our course offering this summer the need for a new type of website; it will combine the functionalities of Google groups, Facebook, ftp server, yousendit.com, email, and academic activities (such as assignment submission and posting of the grades, etc). We had to use various tools to communicate with the students and obtain their submissions. This confused the students on occasion whose feedback suggested the need for such an integrated site. We may develop such an on-line application tool. This may turn out to be useful for many other such hands-on rapid paced activities/courses. (6) We will develop an automation tool to help budding App developers to quickly find APIs, components, frameworks, and Apps, at the appropriate performance level. This will use the semantic web concepts. A research proposal will be submitted to NSF on this in the next one year. (7) We hope that a combination of all these activities will provide us with a steady stream of mobile applications, software tools, and kits that can lead to revenue to sustain our center's growth.

III. Significance

Benefits of Our Approach are listed below:

- User centric and rapid prototyping approach allows the application to be rapidly and reliably fine-tuned/ mapped to different market sectors, while maintaining a user-friendly interface.
- While high-school students were allowed to unleash their creativity, our undergraduate and graduate students also benefited by enhancing their expertise and experience, and thus their employability. They were also trained now to work with the younger entrepreneurs in starting small businesses.
- The young entrepreneurs were involved in evolving user requirements, system specifications, application design, actual prototyping, and in developing a marketing video. We used state-of-the-art languages and tools for all these steps. We also developed automation tools, a priori implemented software and hardware components/subsystems, and an experience/expertise base of our engineering students and their mobile applications, to facilitate the process.
- The final stage of prototyping (debugging and integration) can be very intensive in systems designed bottom-up. However, because we use top-down approaches (coupled with component based bottom-up approaches), prototyping was a relatively painless and fast process.
- Such an infrastructure, if carefully nurtured, has the potential to reduce the cost and the risk faced by VCs (venture capitalists), as they will have good demonstrable prototypes to evaluate, reflect upon, and recommend changes to, which can then be rapidly upgraded.
- The goal is to address the current trends and needs in a timely manner, not 6 to 18 months hence, by which time the trends would have invariably changed. This provides the first mover advantage.
- This innovation infrastructure lowers the barrier for high school entrepreneurs to think 'high-tech'; engineering students now can practice their trade before graduation; more high school students will now want to enroll in science and engineering programs because of this exposure to exciting possibilities and better appreciation of the field.
- Enablement of young entrepreneurial minds to focus attention on commercialization of high-tech products and services, so they can launch high-tech businesses in the longer run, and provide an economic engine for the community. Our on-line survey results (provided in an earlier quarterly report) confirm this.
- Our center benefits by using this project to develop an infrastructure to (1) research and build optimized components, (2) enhance engineering design productivity, (3) develop a seamless system design flow, (4) offer state-of-the-art training courses to our students and help them become more employable, (5) encourage high school students to visualize future opportunities for creating small businesses, and (6) help existing small businesses to succeed in this domain. Some of these efforts will help us with sustenance as a center.

IV. Specific Objectives

We provide an overview of the scheduled activities here. All the scheduled activities have been achieved:

- Summer '09: Prior to this funding, we developed the course and identified a major application ('Android API Search Engine') for prototyping and business plan development.
- Fall '09: (1) Offer an application development course to undergraduate and graduate students, to identify barriers for rapid learning of this high technology, state-of-the-art domain. (2) Prototype an application, viz., the Hangman application. (3) Develop tools for productivity improvement (with the aid of graduate students). We have completed (1) and (2) successfully. We have delayed implementation of one tool for (3) since Android has continued to evolve rapidly. It may benefit from two other existing tools that will improve productivity.
- Spring '10: (1) Upgrade the application development course to emphasize the creativity and UI, and develop more sophisticated applications (relative to fall '09). (2) Offer a component and tool development course to graduate and senior students to develop components/subsystems. (3) Incorporate business planning / commercialization aspects. Learn from these experiences. All these have been completed successfully.
- Summer '10: (1) Recruit high school students (with creative, scientific, or business orientation) and use the course as the incubation center infrastructure to rapidly develop new and robust mobile applications. (2) Use the expertise of our undergraduate and graduate students to help these young entrepreneurs to debug and integrate the software. This would allow the high school students to focus on being creative and application oriented, and yet be conversant and in control of the project direction. (3) Use a panel of engineers and business leaders to select the top student projects. All these have been completed successfully.
- Fall '10: Market Applications developed this summer. Four engineering students are currently supported to tweak the Apps developed this summer, so the university can put these Apps on the market. We expect to put 7 or 8 Apps on the market by December 15, 2010.

V. Methods and Results

The fourth quarter began with the finale of our group's two courses: Android Mobile Component Development and Software-Hardware Co-design with Android. We met on several occasions with members of FAU's School of Communication and Multimedia. After much preparation we were ready for the first week of our Engineering Scholar's Program (ESP) course, Mobile Applications for Google's Android, targeted to high school students. The ESP course lasted for three weeks during which the students completed a significant amount of the work and learned many new skills and concepts. On the last day of the course (7/2/10), the student teams presented their work to a panel of judges. The student teams were ranked based on the judges' scores (70%) and our internal assessment (30 %). Scholarship awards were presented in an award ceremony late in the afternoon the same day. See: <http://www.ceecs.fau.edu/news/nurturing-young-minds> . We also made a presentation on our work at the Mobile Technology Consortium (MTC) in September 2010. See: <http://android.fau.edu/documents/> (note: please scroll down to the end of that web page, to see the MTC presentation).

We have provided the details of the summer course in the fourth quarter report.

Here, we will provide a short summary of the progress made since the completion of the summer '10 course for high school students.

After the conclusion of the Engineering Scholar's Program (ESP) course during the summer of 2010 our FAU Android group began coordinating with Florida Atlantic University's Office of Technology Transfer (OTT). The goal is to commercialize the applications after spending additional development time for quality assurance. Concurrently, our group began participating in the development and teaching two courses at FAU. The first course is called 'Android Projects' where students construct and program autonomous robots that use Arduino microcontrollers. These robots interact among themselves and communicate with Android phones via beacons placed at the edges of the game space. In the second course, entitled 'Software-Hardware Co-design with Android', we continue to teach Android development fundamentals and mentor our engineering students in their efforts to create handheld applications.

The Office of Technology Transfer (OTT) has commissioned our group to refine and improve eight of the nine Android applications that were created during the summer Engineering Scholar's Program course. Four of the eight applications are slated to be released in the month of November while the remaining four intend to be released in the month of December. Charles Norona, Victor Gallego, Brian Chamba, and Alex Andresen are responsible for the development of the said applications. While members of our group were working to achieve the release of these eight applications Professor Ravi Shankar along with the occasional help from the application developers taught the two courses mentioned earlier.

This semester Dr. Ravi Shankar began teaching a new course at FAU entitled 'Android Projects'. It is focused on Android controlled robots. The goal of this course is to develop autonomous robots that will implement the moves in a chess game, as played by two chess players with Android phones. This will be substantially cheaper and more power efficient, as compared to Lego's Mindstorms® Monster Chess Project [See Reference section for the URL for: Monster Chess Video]. The curriculum and projects for this semester's course offering were chosen to help build fundamental knowledge and components necessary to achieve this in later semesters of the course. The overall goal of our efforts in this class and later versions of it is to extend

Android's role in physical computing. A website has been established to help facilitate the dissemination of information involved in this course: <http://robotics.fau.edu>. Joseph Gundel is the TA (teaching assistant) for the course.

Another course that Dr. Ravi Shankar is teaching during the fall 2010 semester at FAU is entitled 'Software-Hardware Co-design with Android'. Much of the curriculum in this course has remained unchanged (refer to the reports of second and third quarters); however, for this semester, we placed less emphasis on concepts such as location-based services, 2-D Graphics, Multimedia, Sensors, and advanced testing with JUnit. Instead, we introduced the Android Native Development Kit (NDK) which allows developers to program at a lower level of abstraction, providing them more efficient control of hardware through the use of the C programming language.

VI. Discussion

Successful completion of our development efforts with the Florida Atlantic University's Office of Technology Transfer (OTT) should enhance the reputation of our university in software and mobile application development. In addition, we will be able to secure funding for our Android group so we can continue to innovate and help high school students and engineering students to start their own small businesses in a highly lucrative area.

With the debut of the Robotics course this semester, our group is furthering the growth of physical computing knowledge as well as contributing to the extension of Android-OS -operated machines to interact with physical computing components. It is our goal to implement a robotic chess game, much like the one created by the author of "Monster Chess" video [See the reference section for the URL on Monster Chess Video]. The difference in our implementation will be that we will try to achieve it for about \$3,000, one-tenth the price of a commercially available Robotic Chess game from Lego [See the reference entitled: 'Monster Chess Article']. Some of the projects that the students have undertaken this semester are on topics such as peer to peer communication, collision avoidance, PID control algorithms, Bluetooth communications (Android-to-Arduino), concurrency programming, vision algorithms, and digital signal processing, all using low-cost, low -power technology solutions. The goal is to make these systems affordable to middle and high schools; further, they will be able to acquire these robots incrementally, so as to build a complete system over a few years. We expect these systems will also be useful in teaching math and physics principles to middle school students, and in teaching engineering principles to high school and college students. See robotics.fau.edu for more information.

The introduction of the Native Development Kit in the Software-Hardware Co-design course will allow our student developers to use the C programming language to reuse existing C code, while improving the performance of the application at the same time. This should allow us to build more powerful and advanced applications in the ensuing years. This will provide our university and our students a distinct advantage. However, our goal is also to continue to fully share details of our applications with others, via our web site (android.fau.edu). Thus, our advantage will be short lived, which will force us to innovate continuously. Thus, as explained above, we have moved on to include remotely controlled robotics as the next area to address. We also plan to

develop games with social impact since such games will take advantage of our process and expertise, while addressing issues of community concern.

VII. Appendices

All of the appendices focus on new and upcoming information. For older documents and information refer to our previous reports from Quarters one, two, and three.

Appendix A: Software-Hardware Codesign Documents

This course used most of the documents which can be found at <http://android.fau.edu/documents/>.

Appendix B: Android Projects Documents

B1. Android Arduino Bluetooth Quick Communication Setup

Objective

To quick be able to communicate to an Arduino Bluetooth board using an Android Modified Bluetooth Chat Application.

Arduino Setup

Follow instructions in <http://www.arduino.cc/en/Guide/ArduinoBT> .

Arduino Echo Code

Create a Sketch called QuickSetup with the following Code:

```
int RESET = 7;

// incoming serial char
char inChar = 0;

void setup()
{
    pinMode(RESET,OUTPUT); /
    Serial.begin(115200); // connect to the serial port
    digitalWrite(RESET, HIGH);
    delay(10);
    digitalWrite(RESET, LOW);
    delay(2000);
    Serial.println("SET BT PAGEMODE 3 2000 1");
    Serial.println("SET BT NAME ARDUINOBT");
    Serial.println("SET BT ROLE 0 f 7d00");
    Serial.println("SET CONTROL ECHO 0");
```

```

Serial.println("SET BT AUTH * 12345");
Serial.println("SET CONTROL ESCAPE - 00 1");
Serial.println("SET CONTROL BAUD 115200,8n1");    //first release 19200
}

void loop ()
{
  if (Serial.available() > 0)
  {
    // get incoming byte:
    inChar = Serial.read();
    //Echo back Char
    Serial.print(inChar);
  }
}

```

Modify Android BluetoothChat Application

Download BluetoothChat program from

<http://developer.android.com/resources/samples/BluetoothChat/index.html> .

The UUID variable needs to be change in the BluetoothChatService.java file in order to communicate with the Arduino Bluetooth board which uses a Serial Profile . In the BluetoothChatService.java file change

```

// Unique UUID for this application
private static final UUID MY_UUID =
UUID.fromString( "fa87c0d0-afac-11de-8a39-0800200c9a66" );

```

To

```

// Unique UUID for this application
private static final UUID MY_UUID =
UUID.fromString( "00001101-0000-1000-8000-00805F9B34FB" );

```

Build the BluetoothChat Application and load it in the Android Phone. Connect to the Arduino Device and send a message. The Arduino will echo back the message.

References:

<http://developer.android.com/reference/android/bluetooth/BluetoothDevice.html>

<http://www.arduino.cc/en/Guide/ArduinoBT>

http://www.sparkfun.com/datasheets/Wireless/Bluetooth/WT32_Datasheet-1.pdf

http://www.sparkfun.com/datasheets/Wireless/Bluetooth/iWRAP3_User_Guide.pdf

B2. Android Installation and Use

Victor Gallego has put together a video list on this. Install the Android IDE only if you are interested in learning more on Android; this is not required for the course - there are enough students in the class with the Android background who can be your partners. If you need help with installation, send email to Victor Gallego, Charles Norona, Brian Chamba, or Joseph Gundel who are in this course. They also spend much of their time in the CSI lab, 417 S&E. You are welcome to meet them there too.

Please head over to <http://android.fau.edu/videos/> to view our Android Installation videos.

Here is the list of videos below:

[Part 01- Download the JDK](#)

[Part 02- Install the JDK](#)

[Part 03- Download the Eclipse IDE](#)

[Part 04- Install Eclipse](#)

[Part 05- Download the Android SDK](#)

[Part 06- Install the SDK Packages](#)

[Part 07- Download and Install the Eclipse ADT Plugin](#)

[Part 08- Create a new Android Virtual Device \(AVD\)](#)

[Part 09- Create a New Android Project](#)

[Part 10- Run your New Android Project](#)

[Part 11- Modify the Output of the Application](#)

[Optional- Add the Tools Directory to your Path Environment Variable](#)

B3. ATMEGA References – 9/16/10

ATMEGA references: 09/16/2010

Motors:

http://www.atmel.com/applications/home_appliances/motor_control.asp?AppID=1040

Security:

http://www.atmel.com/applications/building_automation/fire_security.asp?AppID=1041

Algorithm Builder: An intermediate solution (between C and Assembler):

You can find the Algorithm Builder on AVR freaks web page <http://www.avrfreaks.net/>
and at: (<http://home.tula.net/algrom/english.html/>)

AVR Studio 4: http://www.atmel.com/dyn/products/tools_card.asp?tool_id=2725

B4. More References

1. For practical projects on Arduino, read: Practical Arduino by Jonathan Oxaer and Hugh Blemings. As an example, one of the projects documented there is entitled "Water Flow

Gauge" , pp. 163 - on. It uses an ISR (interrupt service routine) to count the pulses coming from a Hall effect sensor. Other sections that may interest you: DIY Prototype Shielding (pp. 399-400) and Library Creation (pp. 401-409).

2. Joseph Gundel found this Robotic site. He plans to order components and build before he is ready to recommend it. FYI for now: <http://www.trossenrobotics.com/store/p/6331-4WD-Robot-Platform.aspx>

3. Joseph Gunder also found this: A video which uses XBEE Rf module and Arduino as a hoaming device. We might be able to use this for our project.

<http://www.youtube.com/watch?v=cztngp2E94Y&feature=related>

4. Low cost boards: Joseph found this too!
FYI: Arduino Boards on ebay such as

http://cgi.ebay.com/Arduino-Duemilanove-2009-AVR-ATmega168-20PU-USB-board-1-/250687150379?pt=LH_DefaultDomain_0

He purchased one for 18\$

Check out some of these breadboard kits at:

http://www.daqstuff.com/400040_5v_brd_ps_w_ww_.htm

Please let me know if you find interesting info that is useful to others in the class.

5. More from Joseph:

I found where I will buy the NiCAD Batteries from:

<http://www.all-battery.com/aa2300mahnimhrechargeablebatteryflattopformakingpacks.aspx>

They are 1.2V 2000mAh AA batteries. Which are plenty strong. I will put 8 together to make a 12V pack for the motors.

I will use a 9V battery to power the Arduino.

I will use a NiCAD charging circuit which will allow us to charge the battery pack for the motors.

I decided to use this Chassis: <http://www.trossenrobotics.com/p/2WD-arduino-robot-platform.aspx>

B4. References to the Robotics Course

Reference to the Robotics Lego Chess

game: <http://www.youtube.com/watch?v=THGtbBz1sQo>

Another one of the videos which plays through an entire game:

<http://www.youtube.com/watch?v=MAwwKEXn6Mk>

The 3pi robot: <http://www.sparkfun.com/datasheets/Robotics/3pi.pdf>

It is cheaper than the Arduino Bluetooth board. Board alone (\$30) + bluetooth module (\$65) vs Arduino Bluetooth (\$150).

Links: http://www.sparkfun.com/commerce/product_info.php?products_id=666

http://www.sparkfun.com/commerce/product_info.php?products_id=582

http://www.sparkfun.com/commerce/product_info.php?products_id=9211 Looks good includes motors, there are only two unused gpio pins for other sensors. It is 99\$.

Just buying the board(not bluetooth) <http://www.nkcelectronics.com/arduino-diecimila.html> is about \$30. If we buy the sensors and components independantly it might be cheaper.

Another board is:

http://www.sparkfun.com/commerce/product_info.php?products_id=8255 and it is about \$150. There might be a cheaper way to have a bluetooth and arduino using Bluetooth dongle(\$5)

<http://www.circuitsathome.com/mcu/programming/ps3-and-wiimote-game-controllers-on-the-arduino-host-shield-part-1>

Arduino Robot with Bluetooth interface:

http://www.robotshop.com/dfrobotshop-rover-tracked-robot-bluetooth-kit.html?utm_source=google&utm_medium=base&utm_campaign=jos

A good site for Android robot interface. <http://www.markwarner2008.com/android-phone-grows-up-becomes-brain-for-real-robot/>

B5. References – 9/8/10

<http://www.nongnu.org/avr-libc/user-manual/pages.html> - AVR Tool chain
<http://www.arduino.cc/playground/Main/AVR> -- Arduino Assembler hooks
<http://stackoverflow.com/questions/919781/how-can-i-get-an-assembly-language-listing-of-my-arduino-sketches-on-windows> -- Get Assembly language listing
(Easier with a Linux machine/ emulator)
How the code relates to the microcontroller (taking the example of the 'Hello World' of the Embedded world (that is, the LED On-Off-Delay routine:
http://www.urbanhonking.com/ideasfordozens/2009/05/an_tour_of_the_arduino_interna.html
http://linux.die.net/man/3/uint8_t -- definition of the variables
<http://code.google.com/p/arduino/source/browse/branches/servo-library/ServoTimer1/ServoTimer1.cpp> - Arduino code site
wiring.h:
http://www.google.com/codesearch/p?hl=en#thaXon_DSGU/trunk/hardware/arduino/cor/es/arduino/wiring.h&q=wiring.h%20package:http://arduino.googlecode.com&sa=N&cd=1&ct=rc

B6. Websites List

<http://www.nkcelectronics.com/arduino-diecimila.html>
<http://appinventor.googlelabs.com/ode/Ya.html>
<https://www.google.com/accounts/ServiceLogin?service=youngandroid&passive=true&ui=1&continue=http://appinventor.googlelabs.com/ode/&followup=http://appinventor.googlelabs.com/ode/>
http://www.smileymicros.com/index.php?module=pagemaster&PAGE_user_op=view_page&PAGE_id=40&MMN_position=66:66
<http://www.makershed.com/SearchResults.asp?Cat=43>
<http://www.sparkfun.com/commerce/categories.php>
http://www.robotshop.com/dfrobotshop-rover-tracked-robot-bluetooth-kit.html?utm_source=google&utm_medium=base&utm_campaign=jos

References

- Adobe® Photoshop®. Internet site:
<http://www.adobe.com/products/photoshop/family/?promoid=BPDEK>. April 27th, 2010.
- Autodesk® Maya®. Internet site:
<http://usa.autodesk.com/adsk/servlet/pc/index?siteID=123112&id=13577897>.
April 27th, 2010.
- Balsamiq Studios, LLC.. Internet site: <http://www.balsamiq.com/>.
- Brochure of Judges' Profiles. Internet site: http://android.fau.edu/wp-content/uploads/2010/07/JUDGES_BIO_BROCHURE.pdf. July 26th, 2010.
- Camtasia Studio®. Internet site: <http://www.techsmith.com/camtasia/systemreq.asp>.
April 27th, 2010.
- FAU Android Google Group. Internet site:
<http://www.facebook.com/group.php?gid=124108887629373>. July 22nd, 2010.
- FAU Android Group. Internet site: <http://android.fau.edu>. July 22nd, 2010.
- Galoustian, Gisele. "High School Students in FAU's Engineering Scholar Program Compete to Develop Applications for Google's Android Mobile Platform."
Internet article:
<http://www.fau.edu/communications/mediarelations/Releases0610/061018.php>.
July 23rd, 2010.
- "Monster Chess" Article. Internet Press Release: <http://mindstorms.lego.com/en-us/News/ReadMore/Default.aspx?id=193079>. Nov. 10th, 2010.
- "Monster Chess" Video. Internet Video: Youtube:
<http://www.youtube.com/watch?v=THGtbBz1sQo>. Nov. 10th, 2010.
- "Nurturing Young Minds." Internet article: http://www.fau.edu/explore/homepage-stories/2010_07Android.php. July 23rd, 2010.