Rapid Prototyping of Computer Systems
05-540, 05-872, 15-540A, 18-540, 39-648
12 Units
Carnegie Mellon University

Course Syllabus Spring 2005

Lectures:	Monday, Wednesday, 1:30 - 2:50, Hamburg Hall 2224 Wearable Computer Laboratory, Room 2201 Hamburg Hall, 8-6465 Wearable Software Laboratory, Room 2205 Hamburg Hall, 8-3267
Instructors:	Dan Siewiorek 3519 Newell Simon Hall, Ext. 8-2570,dps@cs.cmu.edu Asim Smailagic 1217 Hamburg Hall, Ext. 8-7863, asim@cs.cmu.edu
Secretary:	Mrs. Laura Forsyth, 4207 Wean Hall, Ext. 8-2619 or 8-2570 Electronic mail address - forsyth@ cs.cmu.edu Office hours: 8:30 am to 5:00 PM

Paradigm Shift in Computing

The information processing industry is undergoing a paradigm shift. Commencing in 1960 information processing was concentrated in mainframe computers operated by central staff and accessed by custom-built programs executed in batch mode. By 1970 the invention of the time-sharing operating system allowed users to interact with their information on-line. However, time-sharing systems were still centrally based with most of the computing cycles devoted to information manipulation rather than the human computer interface. With the advent of the personal computer in the early 1980's a substantial portion of the computing power could be dedicated to the single user. New paradigms such as the spread sheet allowed the user to interact with their data on an item-by-item basis looking for patterns and playing "what if" scenarios. Since 1980 technology has been devoted to shrinking the size and weight of personal computers without substantially changing the way users interact with their computing environment. Conventional input/output devices place an ultimate limit on the size and weight of personal computers. Size is limited by the conventional typewriter-like keyboard whose dimensions have not changed substantially for over one hundred years. Both size and weight are limited by displays the size of notebook paper intended to be viewed from several feet. The size of the display places a lower bound on the personal computer's energy consumption and hence weight primarily dictated by the weight of the energy storage devices such as batteries.

The convergence of a variety of technologies makes possible a paradigm shift in information processing. Continued advances in semiconductor technology makes possible high performance microprocessor requiring less power and less space. Decades of research in computer science have provided the technology for hands-off computing using speech and gesturing for input. Miniature heads-up displays weighing less than a few ounces have been recently introduced. Combined with mobile communication technology, it is possible for users to access information anywhere. It is indeed possible to sense a user's position so that the information can be superimposed upon the user's workspace. Mobile computing deal in information rather than programs, becoming tools in the user's environment much like a pencil or a reference book.

Context Aware Computing

Context-aware computing describes the situation where a mobile computer is aware of its user’s state and surroundings, and modifies its behavior based on this information. A user’s context can be quite rich, consisting of attributes such as physical location, physiological state (such as body temperature and heart rate), emotional state (such as angry, distraught, or calm), personal history, daily behavioral patterns, and so on. If a human assistant were given such context, he or she would make decisions in a proactive fashion, anticipating user needs. In making these decisions, the assistant would typically not disturb the user at inopportune moments except in an emergency. Our goal is to enable mobile computers to play an analogous role, exploiting context information to significantly reduce demands on human attention.

The Course Applications: Customer-Centered Car Design and Instrumentation of Science Boat

This is a project-oriented course, which will deal with all four aspects of project development: the application, the artifact, the computer-aided design environment, and the physical prototyping facilities. The class, in conjunction with the instructors and clients, will develop two prototype systems. The projects are an exploration into the evolving areas of context aware computing and real time team collaboration. **Section A** will implement concepts from a customer-centered design car. **Section B** will develop instrumentation for an environmentally friendly scientific vessel for Project Voyager.

Section A: For the past three semester a small group of design and engineering students have been exploring concepts for a customer-centered design for a new generation of automobiles. Following market research, the students have created three user classes and scenarios for trunk storage, seat/driver domain, peripheral device usage while driving (e.g. cell phone, PDA), and technical information presentation/access (especially first time learning such as a hybrid engine car). These scenarios will serve as a point of design departure. These students will serve as clients for Section A.

Section B: Project Voyager (http://www.cmu.edu/voyager) provides an educational experience that enables middle school and high school students to see the application of mathematics and science in the real world. By operating the boats engines, navigating, or taking water samples, students practice what they have learned in the classroom.

Foundations are providing the funding to design a new Voyager that will be environmentally friendly and have the latest multimedia computers, wireless networking, and innovative input/output devices. Last Spring, the Rapid Prototyping class developed concepts such as students on the boat being equipted with a PDA including wireless communications, video camera, two-way audio, and other mechanisms for recording and remote interaction with shore-based students. The boat is being architected and is scheduled for commencement of construction this spring. This project will design instruments for monitoring all forms of emission (chemical, thermal, EM, etc) from both the current boats and the new boat. Data will be automatically entered into the database and appropriate scientific data visualization software. Voyager personnel will serve as clients for Section B.

Upon completion of this course the student will be able to: generate systems specifications from a perceived need; partition functionality between hardware and software; produce interface specifications for a system composed of numerous subsystems; use computer-aided development tools; fabricate, integrate, and debug a hardware/software system; and evaluate the system in the context of an end user application.

Clients will interact throughout the course to provide assistance and advice. A variety of group design productivity tools will be available to class participants to assist students in designing their system.

Course Structure

The course is divided into three major phases (Conceptualization, Detailed Design, and Implementation), each composed of up to several sub phases.

Conceptualization

- Problem Definition. The goal of this sub phase is to define the problem, which is being solved, perform requirements analysis, and evaluate user needs. A variety of brain-storming techniques will be employed to develop a product design definition including attributes such as functionality, cost, performance, technology acquisition, and fabrication techniques.

- Technology Survey. The final shape of a system is often determined by what technology is currently available. A survey of available technology, with special emphasis on input and output devices, will further refine the Product Design Specification. Lessons learned from prior generations of mobile computers will be discussed. New components such as spread spectrum radio and VGA displays will be acquired and interfaced to existing systems to determine the feasibility and complexity of the new technology. Videotapes of current practice as well as discussions with end users will generate interactive scenarios.

- System Architecture Specification. Given the constraints of available technology and the user's computational environment, the architecture for the system will be developed. Topics such as local versus distributed processing, position sensing, computer/human interface, and information updating must be addressed by the selected architecture. Planning will also include interdependencies between the technologies, people, and resources available in the course.

- Subsystem Specification. The system functionality will be partitioned and assigned to hardware and/or software components. Refinement of the Product Design Specification will include attributes of the subsystems including performance, interface, and evaluation criteria.

Detailed Design

- Detailed Design. Each subsystem will be designed and implemented. Design will be supported by contemporary computer-aided design tools. Mini workshops as necessary on relevant tools will provide students a basic introduction to the state of computer-aided design. Human computer interaction studies will be designed in conjunction with mechanical/electronic/software mock-ups to provide data for design decisions.

Implementation

- Implementation. The detailed hardware/software designs will be implemented using both on-campus and off-campus facilities. On-campus physical prototyping facilities will be used when appropriate. The state-of-the-art in rapid prototyping will be presented.

- System Integration. The various hardware and software subsystems must be individually tested and then integrated into a working system. System integration and testing plans will be formulated commencing with the system architecture specification phase. The system will be evaluated through controlled user experiments.

- Methodology Evaluation. As a final phase, the methodology followed in the course will be quantitatively and qualitatively evaluated and modifications suggested.

Tentative Schedule

			Topic	Assignments
Jan.	10	12	**Monday** - Course Overview Wearable Computer Lab Facilities; Brief introduction to projects **Wednesday –** Clients describe projects and give background information. Introduction to Design Methodology, Project Matrix, and supporting tools.	Form teams and meet clients **Wed** in class and from 6 PM to 7 PM
	X	19	**Wednesday** – Further background material. Team Interaction Case Studies Who Am I and What Do I Do;	Generate Scenarios, Functionality from User Interview, Start individual weekly reports and work logs
	24	26	**Monday** – First draft Baseline and Visionary Scenarios. Introduction to how to do Purchasing. **Wednesday –** Technology groups present Product/Feature matrices.	Contact vendors, Product/Feature Matrix, Place orders for initial equipment. Feasibility Experiments. Initial System Specification.
Feb.	31	2	**Monday** –Revised Visionary Scenarios **Wednesday** – Present Architectures: User Interface, Hardware, Software	Final Product/Feature Matrix, Subsystem Specification, Start placing orders
	X	9	Team Presentations **Wednesday** - Section A	Conceptualization and Planning written report and oral presentations, data on Kiva.
	14	16	Detailed Design **Monday** – Section B Team Presentation **Wednesday –** Generate Dependency Graph for rest of semester	Individual feedback to students on progress
	21	23	Detailed Design **Monday** – Revised Scenario reflecting tasks in Dependency Graph **Wednesday** – Meetings to provide students their status and grades	
Mar	28	2	Detailed Design **Monday** **Wednesday** - Design and Tools: The Great Airplane Competition	
	X	X	Spring Break	
	14	16	Detailed Design	
	21	23	Team Presentations **Monday** – Section B **Wednesday** – Section A	Design written report, demonstration, oral presentation, data on

	28	30	Implementation	Kiva Individual student feedback
Apr	4	6	Implementation	
	11	13	System Integration	
	18	20	System Integration	
	25	27	Team Presentations **Monday** – Section B **Wednesday** – Section A	Implementation, Methodology Evaluation, written report, demonstration, oral presentation, data on Kiva

Course Philosophy and Grading

The course is project oriented with a series of phases. In general, each phase will culminate with a written design document and an oral design review. Later phases will also include a demonstration. Students will be divided into project teams whose performance will be graded in terms of classroom participation, written documents, oral presentations, design methodology forms, and demonstrations. Individual feedback will be given in performance reviews at the end of each major presentation. The relative weight for course grading for each of the phases is roughly as follows:

- Technology Feasibility, System/Subsystem Specification - 20%
- Detailed Design - 30%
- Implementation - 30%
- Integration - 20%

Reports Format

In order to capture the relationship between the evolving portions of the design and design process, information will be entered into an on-line repository. Hyper-links will be added between critical decisions. As the design evolves, changes that impact other design decisions will be easier to identify. The documentation will also allow students joining the project later to review the design history. The goal is to provide templates and information that can be used in future courses to improve productivity and minimize design escapes. The report for each phase will follow a standard format and include:

Product
- Product Design Specification
- Attribute Dependency Chart for design parameters
- Design Alternative Evaluation Chart

Design Process
- Inter- and Intra-group dependencies
- Cumulative time line (actual and projected)
- Personnel assignments
- Person-hours effort as a function of task (actual and projected)

Reports will build upon each other and be successive refinements of previous reports. The goal is to document the design evolution as well as the design process.

Project Groups

The class will revolve around project groups. Each group will have the responsibility of designing and implementing one of the scenarios, generate services, or modify the infrastructure, as well as interacting with other groups to ensure compatibility. Groups will consist of about four students, depending on the complexity of the scenario. Groups will be composed of students from multiple disciplines including computer science, design, electrical and computer engineering, human computer interaction, and/or mechanical engineering. A project management council will meet weekly to discuss logistical issues. The council will be composed of the course instructors and a rotating member from each project group. Liaisons between groups will also be utilized. Responsibilities of team members will vary by discipline:

- Computer Science. Select, port, and adapt existing software to unique features of the application, which can include: new device drivers for sensors, input, and secondary storage; new features such as power management, etc. Produce the application data and software.

- Design. Develop application concept. Generate alternative designs for interaction. Assist in implementing application interface.

- Electrical and Computer Engineering. Survey available hardware. Determine what should be purchased and what should be custom designed. Interface with others to minimize modifications to standard software due to hardware design decisions. Purchase components and external manufacturing services as needed.

- Human Computer Interaction. Determine the human-computer interface. Perform the requirements analysis. Design and conduct intermediate experiments to support design decisions. Devise the hardware and software for determining the user's relationship to the displayed information. Conduct a final evaluation of the product and the design methodology.

- Mechanical Engineering. Generate the mechanical structure to support the hardware and selected human interface. Purchase components and external manufacturing services as necessary.

Communications

The major form of communications outside of class will be electronic. Course documentation will be captured in Kiva, kiva.ices.cmu.edu/rpcs05.