User-Centered Interdisciplinary Concurrent System Design

Asim Smailagic, Dan Siewiorek
ICES, Carnegie Mellon University, Pittsburgh, PA

Abstract

As computing devices become more specialized, the user plays an increasingly important role in defining requirements. User expectations for hand-held devices are substantially different from desktop computers. Users expect instantaneous responsiveness as well as intuitive operation. With the advent of rapid design methodologies and rapid fabrication technologies, it is possible to construct fully customized systems in a matter of months. Carnegie Mellon University has developed a User-Centered Interdisciplinary Concurrent System Design Methodology (UICSM) that takes teams of electrical engineers, mechanical engineers, computer scientists, industrial designers, and human computer interaction students that work with an end-user to generate a complete prototype system during a four-month long course. The methodology is web-based and defines intermediary design products that document the evolution of the design. These products are posted on the web so that even remote designers and end-users can participate in the design activities. The design methodology proceeds through three phases: conceptual design, detailed design, and implementation. End-users critique the design at each phase. In addition, simulated and real application tasks provide further focus for design evaluation. The methodology has been used in designing over a dozen wearable computers with diverse applications ranging from inspection and maintenance of heavy transportation vehicles to augmented reality in manufacturing and plant operations. The methodology includes monitoring and evaluation of the design process. While the complexity of the prototype artifacts has increased by over two orders of magnitude, the total design effort has increased by less than a factor of two. This paper describes the methodology and illustrates its effectiveness by describing three recent designs and summarizing their design activities.

1. Introduction

Carnegie Mellon's Wearable Computers project is defining the future for not only computing technologies but also for the use of computers in daily activities. The goal of this project is to develop a new class of computing systems with a small footprint that can be carried or worn by a human and be able to interact with computer-augmented environments. By rapid prototyping of new artifacts and concepts, CMU has established a new paradigm of wearable computers [1]. Sixteen generations of wearable computers have been designed and built over the last five and a half years, with most tested in the field. The user centered, interdisciplinary, concurrent system design methodology [2],[3],[4] has lead to a factor of over 200 increase in the complexity of the artifacts while essentially holding design effort constant. The application domains range from inspection, maintenance, manufacturing, and navigation to on-the-move collaboration, position sensing, global communication, real-time speech recognition and language translation. Since wearable computers represent a new paradigm in computing, there is no consensus on the mechanical/software human computer interface or the capabilities of the electronics. Thus iterative design and user evaluation made possible by the rapid design/prototyping methodology is essential for quick definition of this new class of computers.

In this paper, we describe the methodology and illustrate its effectiveness by describing three recent designs and summarizing these design activities.

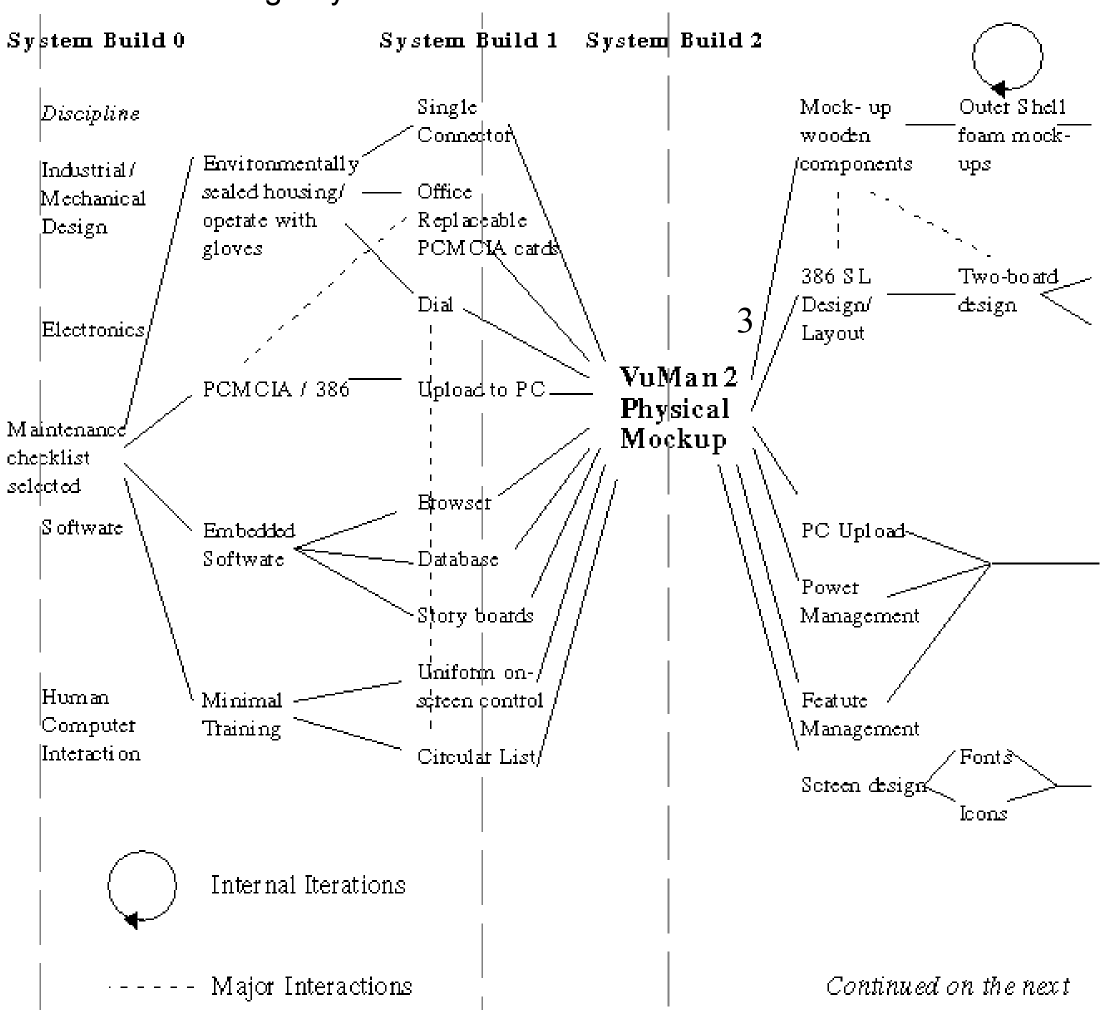
2. User – Centered Interdisciplinary Concurrent System Design Methodology

A User-Centered Interdisciplinary Concurrent System Design Methodology (UICSM) [2], based upon user - centered design and rapid prototyping, has been applied to the design of wearable computers. Based on user interviews, and observation of their operations, baseline scenarios are created for the current procedures. Visionary scenarios identify opportunities for technology injection. User feedback on scenarios and storyboards become input to the conceptual design phase. Designers alternate between the abstract and the concrete; preliminary sketches are evaluated, new ideas emerge, and more precise drawings are generated. This iterative process continues with soft mock-ups, appearance sketches, computer and machine shop prototypes, until finally the product is fabricated.

As a result of UICSM, we have achieved a four month design cycle for each new generation of wearable computers. The cycle time of the new products is ideally suited to the academic semester. As depicted in Figure 1, student designers initially visit the user site for a walkthrough of the intended application. A second visit after a month of design, ending the conceptual phase, elicits responses to story boards of the use of the artifact and the information content on the computer screen. After the second month a software mock-up of the system running on a previous generation wearable computer is evaluated in the end-user's application, representing the results of the detail design phase. During the third month, implementation takes place and a prototype of the system receives a further user critique. The final system is delivered after the fourth month for field trial evaluation.

The goal of the UICSM is to allow as much concurrency as possible in the design process. The design cycle includes monthly "builds," an evolving system integration demonstration that solicits end-user feedback each month. The System Build phases and the iterative nature of user-centered design in a four month design cycle are also represented in Figure 2. Figure 2 illustrates the concurent activities in four disciplines for the system depicted in Figure 1. The application supported by VuMan 3 is Vehicle Inspection. The final design of VuMan 3 is described in Section 4.1.

Figure 1. Four Month Design Cycle



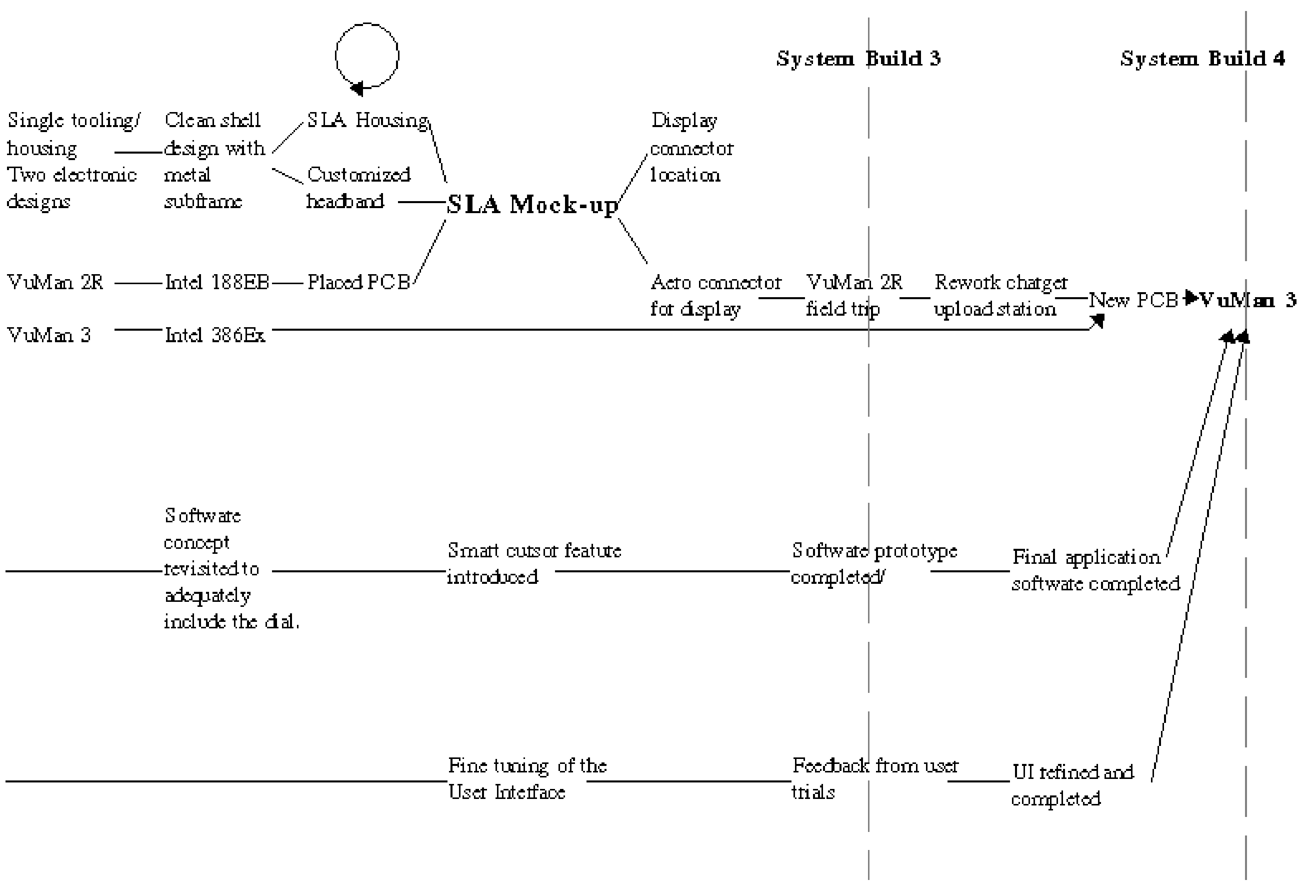
Continued on the next page.

Figure 2. Concurrent Design Activities and System

Stage 0 System Build: The first visit to the users site generated the major attributes and design requirements imposed by the heavy military vehicle maintenance application, Figure 2. The first step in the inspection process, the Limited Technical Inspection (LTI), was chosen for the application. The LTI is a checklist of over 600 items, in a 50-page paper document. The selection of maintenance checklist application represents the primary design decision. We observed the maintenance process, the operating environment, the tools, and supporting activities. Maintenance workers wore gloves and operated in confined, environmentally harsh spaces in both subdued lighting and bright sun light. The mechanics worked in dust, fuel, and corrosive chemicals. Mechanics review procedures in manuals stored at a central repository before returning to the vehicle, occasionally with copies of critical manual pages. Further task analysis yielded the first design constraints - the environmental sealed housing, an input device that could be operated with gloves, a singe integrated housing whose interface could be operated independent of orientation.

Stage 1 System Build: During the second phase we configured the new product to produce the first concept of the total system. A second visit after a month of design elicited responses to story boards of the use of the artifact and the information content on the computer screen. The story board mock-up of the application includes a "slide show" of the application on a previous generation wearable computer, and represents the first "build" phase in the evolution. Estimates of the processing and data storage requirements indicated that a 386 processor and PCMCIA memory cards would provide enough processing power and memory capacity. The data collected during a vehicle inspection would be uploaded to a logistic computer. Another goal was less than five minutes training time. The user interface adopted uniform on-screen control icons to simplify screen manipulation.

Stage 2 System Build: This stage involves several physical mock-ups as well as a software prototype on an existing platform, in this case VuMan 2. On the third visit to the maintenance base we demonstrated the prototype application software, and demonstrated various forms for the input dial Figure 3. Based upon user feedback design proceeded with making the wooden mock-up to visualize and configure the housing. After exploring the implications on the housing configuration it was decided to decrease the footprint of the design by increasing its thickness with a two board design, the main processor board, Figure 4, and a PCMCIA controller board. The PCMCIA Card Controller chip (82365SL) minimized chip count of the interface between the processor ISA bus and the PC Card socket. The SLA (stereolithography) mock-up model of the housing was created at the end of this stage.



Stage 3 System Build: To reduce the risk of switching to a more complex electronic component set from a less complex a two phase approach was adopted. The first system, VuMan 2R, would use an existing processor from VuMan 2, the Intel 188EB; the second system, VuMan 3, would incorporate an Intel 386 processor. The 386 processor board would cover the batteries whereas the 188 would have cut-outs for the batteries resulting in a thinner housing. The housing for both designs would be the same, the only difference was that the 386 housing would have an increased vertical height. Only one set of housing tools was required.

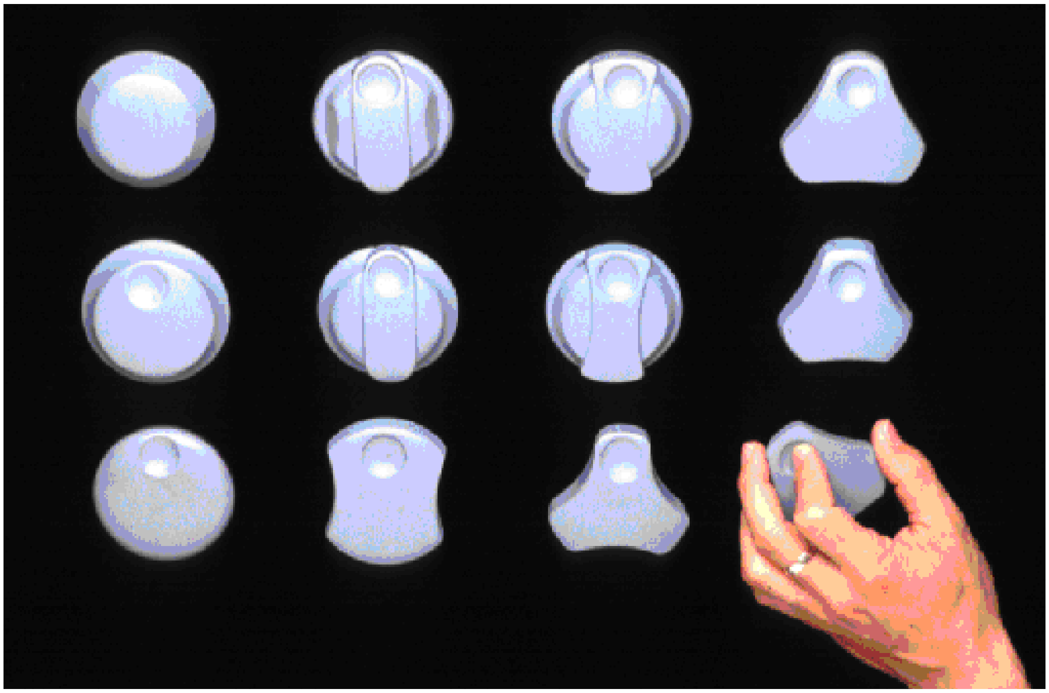


Figure 3. Different Forms for the Input Dial for VuMan 3

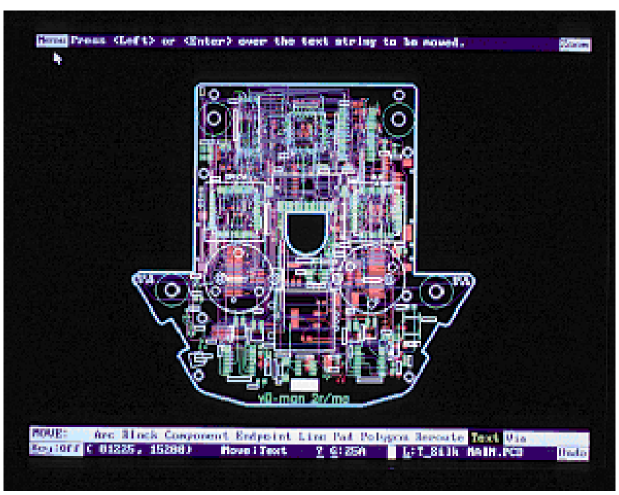


Figure 4. Main Processor Board Design

Figure 5. VuMan 3 Wearable Computer

Stage 4 System Build: The final wearable computer, Figure 5, was a 5"x6.25"x2" unit weighing less then two pounds, including a rotary dial input device integrated with an environmental sealed housing, Private Eye display with a customized headband, a smart docking station which monitors the use of the NiCd rechargeable batteries and also acts as a communication link to a host computer system.

3. Web - Based Design Methodology

The design methodology described in this paper is web-based and defines intermediary design products that document the evolution of the design. These products are posted on the web so that even remote designers and end-users

Project Matrix			
Discipline	Product Development Phases		
	Conceptual Design	Detailed Design	Implementation
Hardware (HW)	1 Review field data/refine HCI Problem scenario 2 Select/refine HCI Target technologies 3 Review/refinel HCI Visionary scenario 4 HW Product/feature matrices 4 HW Feasibility studies 5 Input to PM Design Decision tracking form (HW selection criteria and choices) 6 HW architecture	9 Input to HCI Design scenario 10 Add to Task / Issue Tracking form 10 Add Resolutions to Task/Issue Tracking form 11 Status list HW tasks and issues 12 Provide input to schedule 13 Resolve issues 13 Perform unit HW implementation 14 HW design phase summary 14 User evaluation and feedback plan	17 Updates to status list of HW tasks and issues 19 Input to HCI Demo Script 20 Input to PM Integration Tree 21 Integrate HW components 22 Dry run of demo and testing
Software (SW)	1 Review field data/refine HCI Problem scenario 2 Select/refine HCI Target technologies 3 Review/refine HCI Visionary scenario 4 SW Product/feature matrices 4 SW Feasibility studies 5 Input to PM Design Decision tracking form (SW selection criteria and choices) 6 SW architecture	9 Input to HCI Design scenario 10 Add to Task / Issue Tracking form 10 Add resolutions to Task/Issue Tracking form 11 Status list SW design tasks/issues 12 Input to schedule 13 Resolve issues 13 Perform unit SW implementation 14 SW design phase summary 14 User evaluation and feedback plan	18 Updates to status list of SW tasks and issues 19 Input to HCI Demo Script 20 Input to PM Integration Tree 21 Integrate SW components 22 Dry run of demo and testing 23 Archive and document source and object
Mechanical / Industrial (MEI)	1 Review field data/refine HCI Problem scenario 2 Select/refine HCI Target technologies 3 Review/refine HCI Visionary scenario 4 MEI Product/feature matrices 4 MEI Feasibility studies 5 Input to PM Design	9 Input to HCI Design scenario 10 Add to Task / Issue Tracking form 10 Add resolutions to Task/Issue Tracking form 11 Status list MEI design tasks/issues 12 Input to schedule 13 Resolve issues	18 Updates to status list of MEI tasks and issues 19 Input to HCI Demo Script 20 Input to PM Integration Tree 21 Integrate MEI components 22 Dry run of demo and testing

Table 1. Project Matrix

Human Computer Interface (HCI)	1 Field evaluation reports and data 1 Problem scenario 2 Target technologies 3 Visionary scenario 4 HCI Feasibility studies **5 Review product/feature matrices** **5 Review feasibility studies** 5 Refined solution scenario 6 Initial user interface concepts	9 Input to HCI Design scenario 10 Add to Task / Issue Tracking form 10 Add resolutions to Task/Issue Tracking form **11 Status list of UI design tasks/issues** **12 Input to schedule** **13 Resolve issues** **13 Perform unit UI implementation** **14 Design phase summary** **14 User evaluation and feedback plan** **15 Coordinate user evaluation and prepare feedback report**	**18 Updates to status list of HW tasks and issues** **19 Produce Demo Script** **20 Input to PM Integration Tree** **21 Integrate user interfaces** **22 Dry run of demo and testing** **23 Archive and document source and object**
Cross- Functional Groups	1 Review field data/refine HCI Problem scenario 2 Review/refine HCI Target technologies 3 Review/refine HCI Visionary scenario 4 Product/feature matrices 4 Feasibility studies **5 Input to PM Design Decision tracking form (selection criteria and choices)** **6 High level design**	9 Subsystem interface specifications 10 Add to Task / Issue Tracking form 10 Add resolutions to Task/Issue Tracking form **11 Status list subsystem tasks/issues** **12 Input to schedule** **13 Resolve issues** **13 Perform unit subsystem implementation** **14 Design phase summary** **14 User evaluation and feedback plan**	**18 Updates to status list of subsystem tasks and issues** **19 Input to HCI Demo Script** **20 Input to PM Integration Tree** **21 Integrate subsystem** **22 Dry run of demo and testing**
Project Management (PM) Group Leaders	**Bi-weekly update of form data** **Develop work breakdown/schedule** **Phase 1 Task dependency graph**	**Bi-weekly update of form data** **Update work breakdown/schedule** **Phase 2 Task dependency graph**	**Bi-weekly update of form data** **Update work breakdown/schedule** **Phase 3 Task dependency graph** **20 Produce integration tree.**
Team Products Link to group membership and meeting pages	2 Requirement Table 6 Requirement - Feature table 7 Product design specification **8 Presentation slides** **8 Conceptual Design**	15 Product design specification **16 Presentation slides** **16 Detailed Design Phase Report**	21 Product design specification **22 Presentation Slides** **23 Final Report**

can participate in the design activities. Table 1 depicts the Project Matrix used in the course. The design methodology proceeds through three phases: conceptual design, detailed design, and implementation. The numbers represent concurrent activities in the various disciplines. At different times, different disciplines initiate activities with other disciplines contributing to the results. For example, the HCI group performs a field evaluation and produces the problem scenario which is reviewed and refined by the other groups (Step 1 in Table 1). Individual disciplines are responsible for generating technology specific product/feature matrices (Step 4 in Table 1) for the target technologies identified (Step 2) to support the visionary scenario (Step 3). Table 2 is an example product feature matrix of speech recognition systems, identifying features that were studied

Company	Product	OS	Processor	Memory	Space	Price	Features
Dragon Systems	Naturally Speaking (personal, deluxe versions)	Win95, WinNT	Pentium 133 Mhz Also requires sound card and CD ROM.	48 MB	100 MB	$ 129 (personal) $ 695 (deluxe)	230,000 Vocabulary (personal version). Additional features in deluxe version. Dictate into any application. Voice control of mouse pointer. Multiple Users. Ability to write macros. Playback of recorded speech.
Dragon Systems	Dragon Dictate 3.0	Win95, WinNT	Pentium 133 Mhz Also requires sound card and CD ROM.	32 MB	36 MB	$149	Discrete speech. Dictation playback. Text-to-speech conversion.
IBM	Via Voice Gold	Win95, WinNT	Pentium 150Mhz Also requires Sound Blaster and CD ROM drive.	32 MB (Win95) 42 MB (WinNT)	125 MB	$99	Dicate into any application. Macros. 22,000 Vocabulary (can expand to 64,000) Includes a noise cancellation headset microphone.
IBM	Simply Speaking Gold	Win95, WinNT	Pentium 100Mhz Also requires Sound Blaster and CD ROM drive.	32 MB	36 MB	$99	Discrete speech. Dictation playback. Text-to-speech conversion.
Philips	SpeechMagic	Win3.1	486 66 Mhz Also requires Philips LFH 6210 Accelerator Board.	16 MB	> 500 MB	$???	64,000 word vocabulary. Speaker adaptation. Provides a correction editor, editing and playback of recordings, vocabulary manager for adding new words. Windows DDE support and native API provided for integration.

Table 2. Product Feature Matrix for Speech Recognition Systems Indicating Functions and Hardware / Software Requirements

and evaluated in Step 4. The Conceptual Design Phase concludes with discipline specific architecture definition (Step 5) The other two phases proceed in a similar manner. Cross functional teams insure consistency between disciplines. Group leaders form a Project Management Team responsible for execution of the methodology. Each phase culminates in web products, a written report, and an oral presentation produced by the entire group. These activities are represented at the bottom of Table 1.

4. Taxonomy of Wearable Computers

Wearable / Mobile computers can engage the senses of an individual to enhance the performance of that individual as well as a team that the individual is associated with. We treat collaboration as a type of enhancement of the human senses. In that respect, three examples of CMU wearable computers can be classified by the taxonomy shown in Figure 6. VuMan 3 [5] represents a stand-alone wearable computer performing data recording functions. The C-130 employs one - to - one collaboration in the form of a help desk. MoCCA [8] involves a group in collaboration. The following subsections summarize the final design for these three examples.

Number of Collaborators	Taxonomy	Wearable Computer
0	Stand alone (Data Recorder)	VuMan 3
1	One – to – One Help Desk	C130
N	Group Collaboration	MoCCA

Figure 6. Taxonomy of Example CMU Wearable Computers with Respect to Collaboration

4.1 Stand Alone Computer : VuMan 3

VuMan 3 design represents a successful experience in electronics / mechanical / software co-design, in particular software / mechanical integration, by coupling a novel customized input device, the rotary dial, and the application software. VuMan 3 functions as a data recorder and facilitates the filling out of forms. It was designed for making the Limited Technical Inspection (LTI) of amphibious vehicles for the U.S. Marines at Camp Pendleton, CA more efficient. The result was a new user interface paradigm: circular input, circular visualization.

The LTI checklist consists of a number of sections, with about one hundred items in each section. The users can manually progress from item to item by using the dial to select "next item", or "next field." Selectable items are represented as bold face in Figure 7. During the Second System Build Stage, users indicated that dial motion could be reduced by taking advantage of the fact that on average over 80% of the items on a LTI are serviceable. The Smart Cursor feature represents built-in intelligence in the user interface, and was designed to help automate navigation of the checklist. The Smart Cursor includes:

- An input pattern recognizer, which keeps track of what fields the user selects on a given screen, forming a "working set". If the working set remains the same over two or more screens, the Smart Cursor starts moving the cursor automatically to the fields in the working set. In essence, this is a macro recorder that runs continuously during the user's work session, and uses a heuristic about when to repeat recorded inputs.

- A domain-specific heuristic, developed through studies of how users usually navigate through LTI hypertext documents (e.g. their behavior in the presence of multiple options). A high-level navigation pattern was found, which the

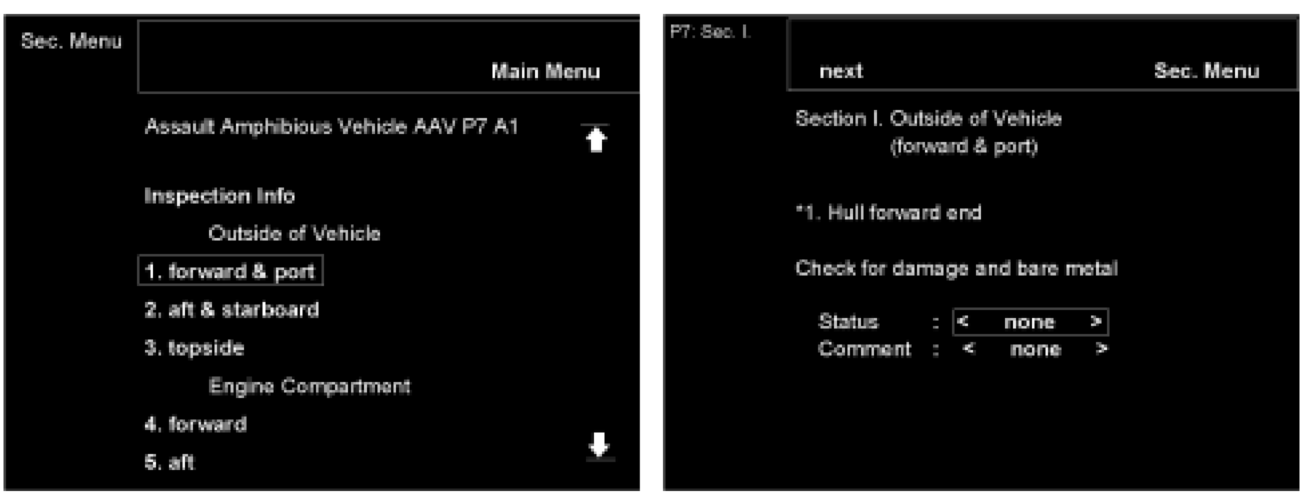


Figure 7. LTI Information Screens

input pattern recognizer could not identify. The knowledge about this high-level pattern was encoded in a navigation template. The system then uses a heuristic to decide when to apply this navigation template. This results in skipping to the next status box (top of right screen, Figure 7), when selecting the serviceable option.

A dial is simpler and inherently less expensive than a trackball because it has only one degree of freedom (and hence one transducer, making it less expensive to manufacture). A trackball is a continuous 2D input device, more complex to build and operate (manual dexterity). A dial requires a rethinking of the user interface to map a one-dimensional input device to a two-dimensional selection surface. Information displaying software includes a tight integration between input device and software. Options to be selected can be logically arranged in a circular list. An option is highlighted. Rotation of the dial one position clockwise changes the highlighted option to the next one clockwise in the circular list. The same applies for counterclockwise. There may be more, less, or equal number of positions on the dial than in the circular list. Depression of one or more buttons performs the function specified by the highlighted option, enters the highlighted information into a database, provides auxiliary information, selects a hypertext link, selects an option on a World Wide Web page, etc. This can be applied to any application in which the contents of a single screen can be represented as a circular list of selectable items. Each element of the list can be linked to another screen that is invoked when the list element is selected. This includes checklists, as well as hypertext and labeled diagrams. From a visual perspective, selectable items can be placed around the outside of the screen and can be navigated in a circular fashion. Thus the screen look and feel corresponds to the input device and the end user feels as if they are dealing with a single, unified device. Application software was modified to reflect the dial design decisions and the final result was a new user interface paradigm: circular input, circular visualization. The rotational input means that it has discrete, tactilely discernible positions. The dial can be operated while wearing gloves, and the interface operation is unambiguous and the same no matter where the dial is worn on the body.

4.2 One - to - One Help Desk : C-130

A multimedia collaboration system with head-mounted display, video camera, and wireless communications provides access to electronic maintenance manuals and remote access to a human help desk expert on the C-130 flight line. Use of wearable comptuers for collaboration has been reported in [6],[7]. The C-130 project was designed to use collaboration to facilitate training. Inexperienced users were being trained to perform a cockpit inspection and the trainers were located in the more spacious cargo hold of the C-130. There is no reason why the trainers need to be located in the aircraft and the ultimate goal of this experiment was to support more students with the same number of trainers. The student loads the inspection procedures and performs the inspection. The C-130 system used the dial as an input device. A desktop system managed the normal job order process and was used by the instructors in the C-130 project to observe the student's behavior. The instructor looks over the shoulder (through a small video camera attached to the top of the student's head mounted display) and advices when problems occur. The advice includes an instructor controlled cursor for indicating areas on the video image which is being shared through a whiteboard. The instructor manages the sharing session and whiteboard. The students' use of of the white board is limited to observation. The functions that were performed by the technicians were determined by observation which, in turn, generated the functionality made available through the user interface. The wearable user has two main functions: navigating through the checklist and initiating collaboration. All navigational links are either through the natural sequence of the checklist or through a simple menu. The novice uses a dial to sequence through active regions of the screen. Figure 8 illustrates an example of the user interface that supports a checklist application. The student is shown in Figure 9.

The interface is organized as a hierarchy with a sequential list of inspection steps embedded in the hierarchy. Each screen gives the ability to go to the next or previous step of the inspection. Failures can be indicated within the instructions.

The collaboration is managed by the remote expert from a workstation. A collaboration will consists of the novice verbally stating something about a problem and the expert indicating with the cursor and describing verbally the actions to be taken. Since the expert is at a desktop, moving a cursor to any position on the field of view is not a problem.

The C-130 application software was constructed using World Wide Web (WWW) browsers and took advantage of the hypertext linking facility supported by that software. Special helper applications were written in Java to provide collaborative drawing services. The database component consisted of a collection of databases both newly created and preexisting. The C-130 project had databases that contained checklist information. The middleware included a WWW server that provided interfaces to the databases and generation of HTML with the information from the databases included. The middleware also included the communication aspects of the collaboration software. The electronics provided sufficient resources so that a full function operating system (Windows 95) could be used on the wearable platform. The database and middleware components of the software were resident on the desktop.

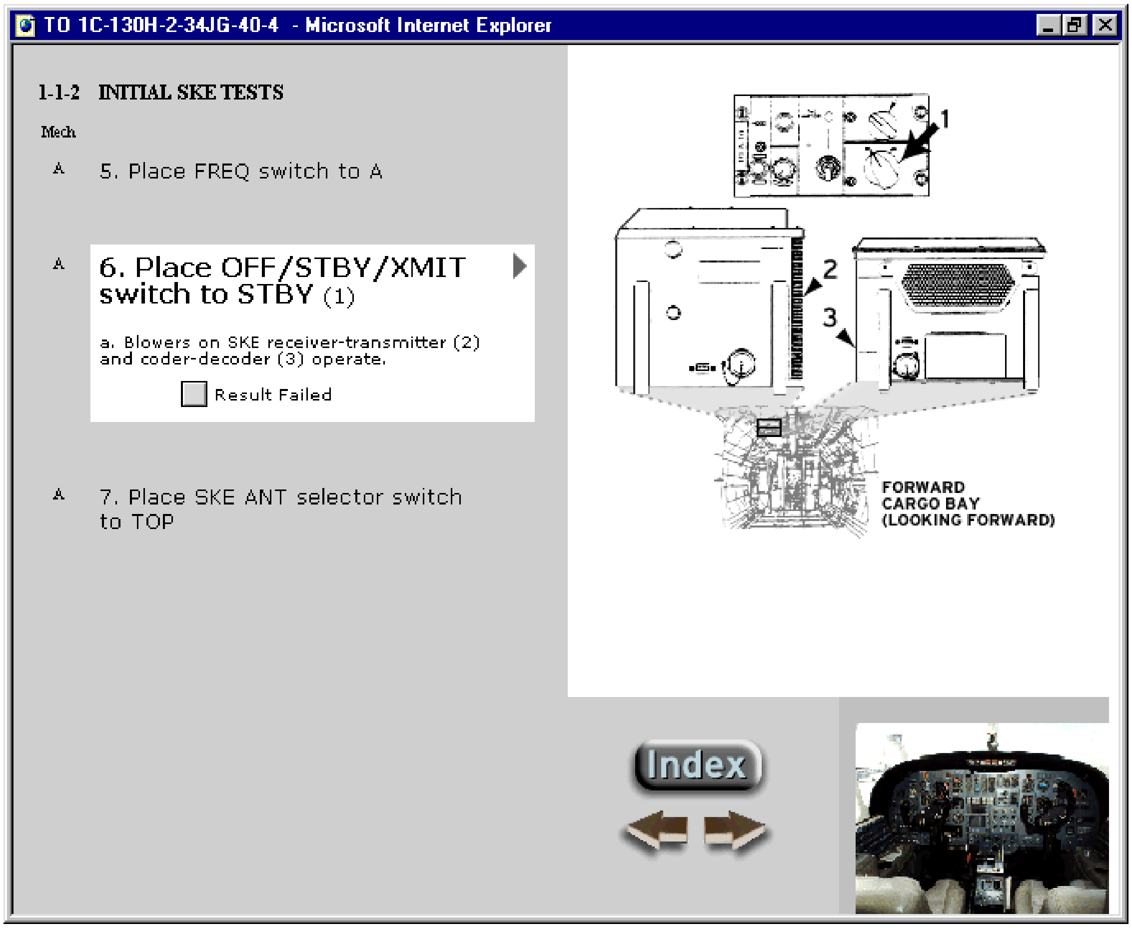


Figure 8. User Interface Screen

Figure 9. Interior Inspection of a C-130 Cockpit

Microsoft's Internet Explorer was used for the C-130 project because it was easy to integrate the dial as an input device. Internet Explorer allows Tab and Shift-Tab as mechanisms for navigating through the links on a page. A clockwise rotation of the dial was mapped into the Tab and counterclockwise into the Shift-Tab, and consequently, the browser could be used without modification. The software design was chosen primarily to enable the utilization of commercially available components both for display and communication. In this case, it was the software that constrained the electronics to be sufficiently powerful to support a fully functional operating system that in turn, supported the WWW software.

The C-130 wearable computer was designed by composition, using mainly off the shelf components integrated into the TIA-P and TIA-0 wearable computers. TIA-P is a commercially available system, developed by CMU, incorporating a 133 MHz 586 processor, 32MB DRAM, 2 GB IDE Disk, full-duplex sound chip, and spread spectrum radio (2Mbps, 2.4 GHz) in a ruggedized, hand-held, pen-based system designed to support speech translation applications. The TIA-P is shown in Figure 10. For this application, the TIA-P was adapted so that a dial provided all of the necessary input capability except for the video camera and associated video capture card.

The TIA-P has been demonstrated with the Dragon speech translation system in several foreign countries. TIA-P has supported speech translation demonstrations, human intelligence data collection, and experimentation with the use of electronic maintenance manuals for F-16 maintenance.

TIA-0, show in Figure 11, is a small form factor system using the electronics of TIA-P. The entire system including batteries weighs less than three pounds and can be mission-configured for sparse and no communications infrastructures. A spread-spectrum radio and small electronic disk drive provide communications and storage in the case of sparse communications infrastructure whereas a large disk drive provides self-contained stand-alone operation when there is no communication infrastructure. A full duplex sound chip supports speech recognition. TIA-0 is equivalent to a Pentium workstation in a softball sized packaging. The very sophisticated housing includes an embedded joypad as an alternative input device to speech. The TIA-P has a handheld screen integrated into computer, while a modular design of the TIA-0 enables the use of a head - mounted display, or a handheld screen. The TIA-0 is also commercially available.

Figure 10. TIA-P

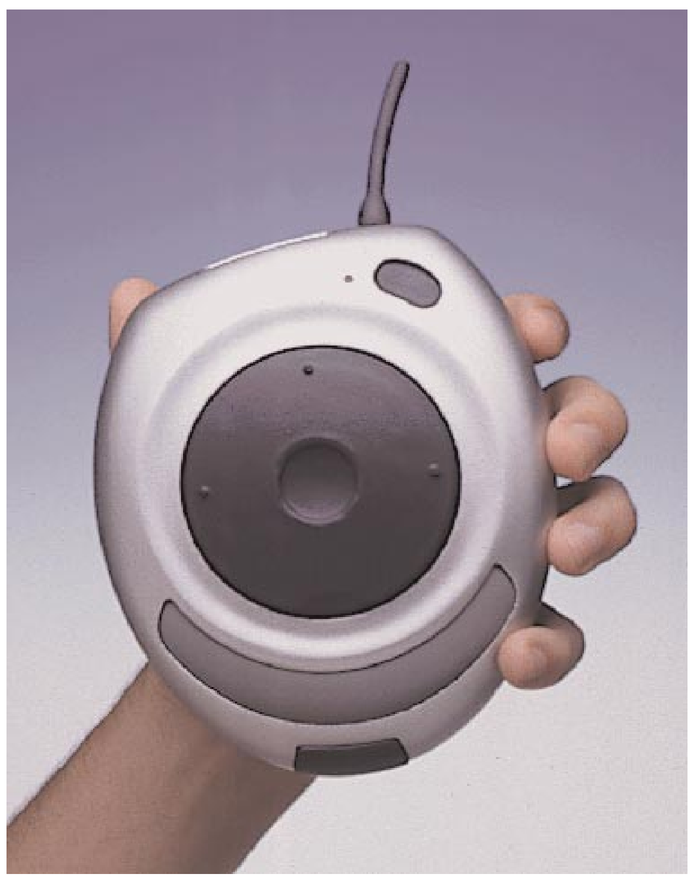


Figure 11. TIA-0

4.3 Group Collaboration : MoCCA

The Mobile Communication and Computing Architecture (MoCCA) was designed to support a group of geographically distributed field service engineers (FSE). The FSEs spend up to 30% to 40% of their time in a car driving to customer sites. Half of what they service is third party equipment for which they may not have written documentation. The challenge was to provide a system that allowed the FSE's to access information and advice from other FSEs while on customer sites and while commuting between sites. Synchronous and asynchronous collaboration are supported for both voice and digitized information.

4.3.1 Conflicting Design Requirements

An additional challenge arose from user interviews which suggested that the FSEs desired all of the functionality of a laptop computer including a larger color display with an operational cycle of at least eight hours. In addition the system should be very light, preferably less than one pound in weight, and require access to several Legacy databases that existed on different corporate computing systems. Further discussions with the FSEs indicated that the most frequently used databases were textually-oriented. Only on rare occasions was access to graphical databases required. However, when required, they were absolutely necessary. To address this set of conflicting requirements, the electronics designers created a novel architecture that combined a light weight alphanumeric satellite computer with the high functionality of a base unit which is included in the FSEs tool kit.

4.3.2 Architecture

MoCCA consists of the following units (Figure 12):

- A base unit, about the size of a small laptop computer, which is connected to a remote server wirelessly through a CDPD connection. Located in the FSE's tool kit, the base unit has a large color display for viewing schematics.
- A cellular phone is tethered to the base unit and communicates wirelessly with the local cellular provider.
- A hand carried satellite unit which is in communications with the stationary base unit. The satellite unit displays the contents of the base unit screen. The satellite unit weighs less than 0.75 pounds.
- A microphone and headset combination which is wireless linked to the cellular phone.

The software architecture uses a thin client approach to minimize the amount of software on the base unit by exploiting web-browsing technology and wireless CDPD Internet connection to communicate with a server. The satellite unit is not running the browser; it is merely displaying whatever is currently on the base unit display.

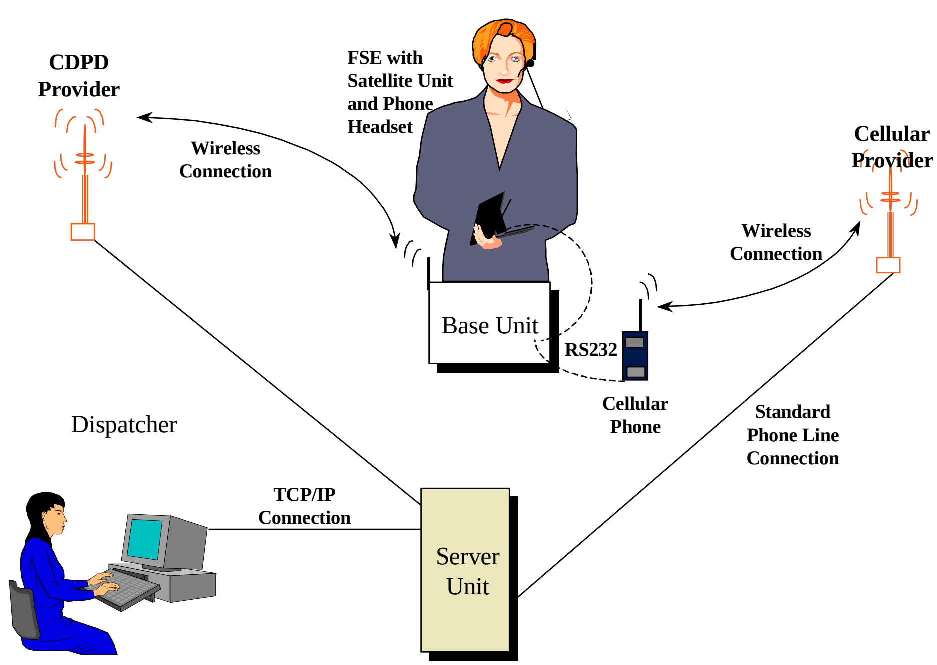


Figure 12. MoCCA System Architecture

A summary of the integrated user interface software is presented in Figure 13. The Field Service Engineer Call List is shown as the central screen and all other screens can be accessed following hypertext links or the screen buttons. From the Call List screen, the user can select a primary function to be performed, such as database query (Call List) or change. Depending on the primary function selected, subsequent secondary screens will display more detailed information. In the case there are changes in the database, the secondary screens allow the user to enter new information.

There are six buttons that appear on all screens of the user interface. The Bboard button provides access to a phone-based voice bulletin board where FSEs may asynchronously collaborate to solve problems. The Calls button accesses the summary of active field service calls for the engineer. The Phone button invokes an auto-dialer keypad. The FSE button brings up a directory of FSEs, and the Availability button shows their current status. The Pager button accesses the list of current Pager messages. The FSE directory, Pager message list, Calls list, and Availability form are all web pages, generated automatically from various field service databases.

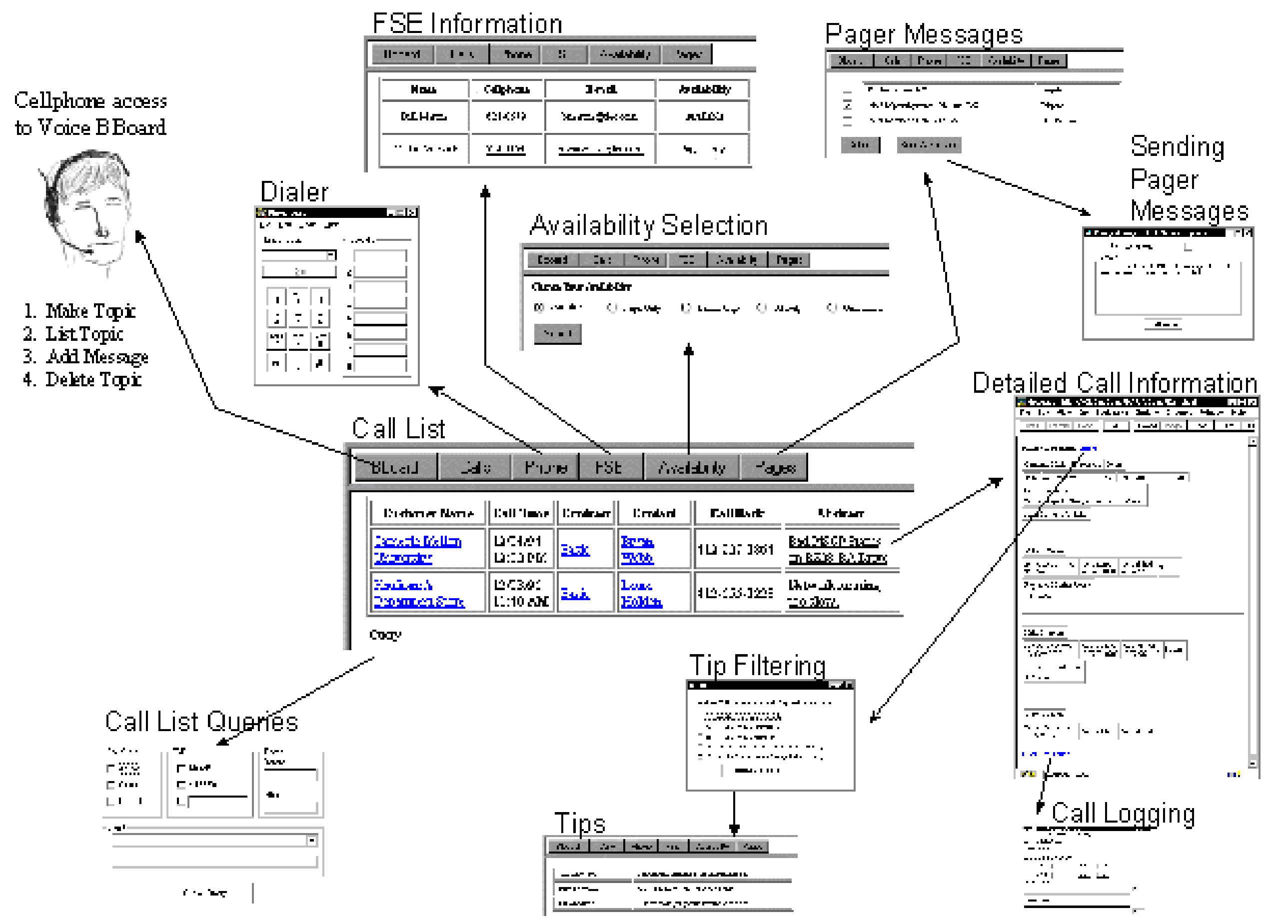


Figure 13. Summary of the Integrated User Interface Software

The FSE checks his current status and the status of other FSEs by clicking on the "FSE Information" button. This leads to a web page (Figure 14a), generated from the MoCCA database, listing the names of all the FSEs, their cell phone numbers, their e-mail addresses, and their current status. The FSE will change his status from "Off Duty" to "Available" to let others know that he can be reached by cell phone. This is done by clicking on the "Availability" button at the top of the screen, leading to the Availability Selection page, Figure 13. Clicking on the "Available" button leads back to the FSE Information page. To check the list of calls assigned to the FSE, he clicks on the "Calls" button at the top of the screen, and brings up the "Call List" screen, Figure 14b. The page is assembled from a database query of the calls assigned to the FSE. It filters the list of all calls to only display those with an "Open" status. The table lists each call with the customer name, the time the call was received, the service contract, the contact person, their phone number, and a short description of the problem. Each underlined entry leads to further information about that field. If the FSE wants to review the assigned calls, and the previous problems at a site, he clicks on the "Call Query" button, which brings up the "Call List Queries" page. To get more details on a particular call, the FSE clicks on the description of the problem, showing the "Detailed Call Information" list. By clicking on the "Service Log" button, the FSE can review the Service Log entries associated with that call, leading

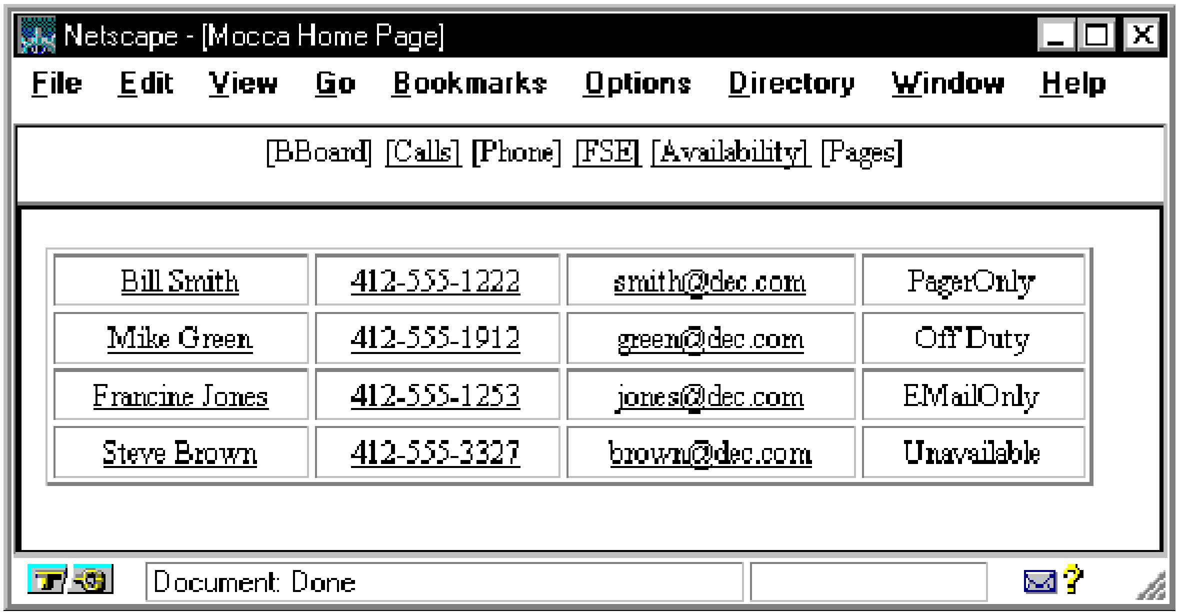


Figure 14a. Sample Screen Image for FSE Information

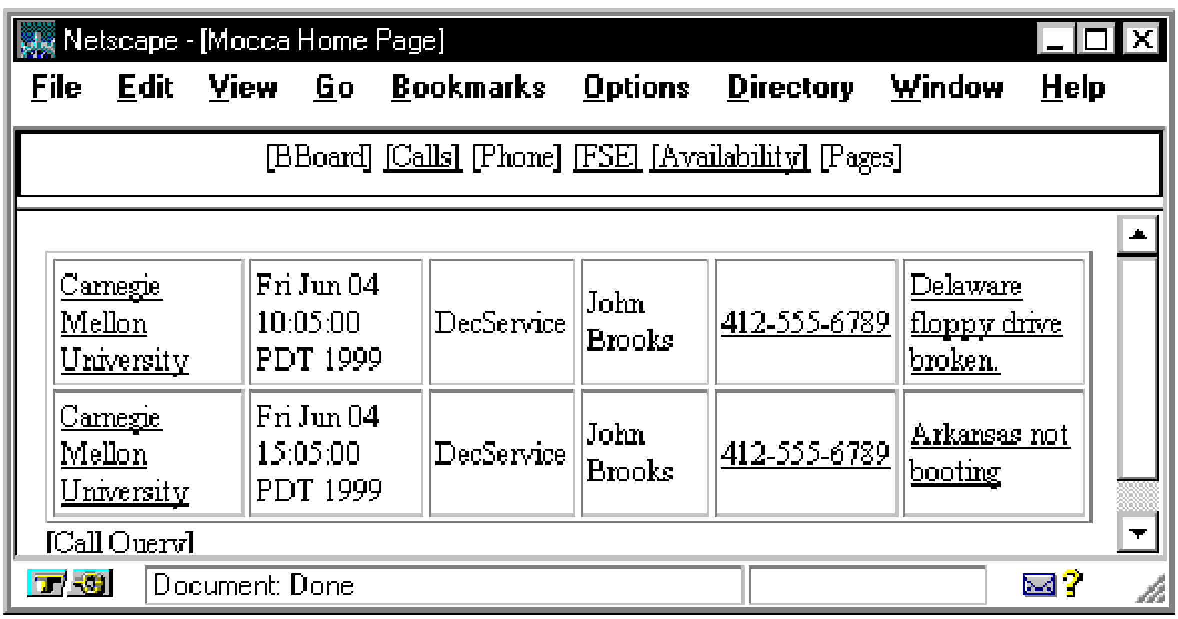


Figure 14b. Sample Screen Image for Call List Pages

to the "Call Logging" list. There is a "Tip" recorded with information on most of the parts to be replaced , and the FSE can examine its content. After reviewing the call history, he can look at the current "Call List" again.

The satellite unit enabled FSEs to wear a very small, lightweight computer with display. An important design decision was to use the same user interface as the wearable base station. The Cassiopeia satellite unit has a screen size of 2.5" x 4.75" and a resolution of 480x240 pixels while the base station's resolution is 640x480 pixels. Only a 480x240 pixel window was used for the screens shown in Figure 13. The software used to send information from the base unit display to the satellite unit is PC Anywhere under Windows CE.

Figure 15 depicts the base unit components and the satellite unit hardware. The base unit includes a 586 133 MHz processor, running the Windows 95 operating system. There are two PCMCIA slots on the base unit: one is occupied by the AirCard CDPD/Modem, and the other by the modem dialing to the cell phone. RF transceivers link the cell phone and the headset.

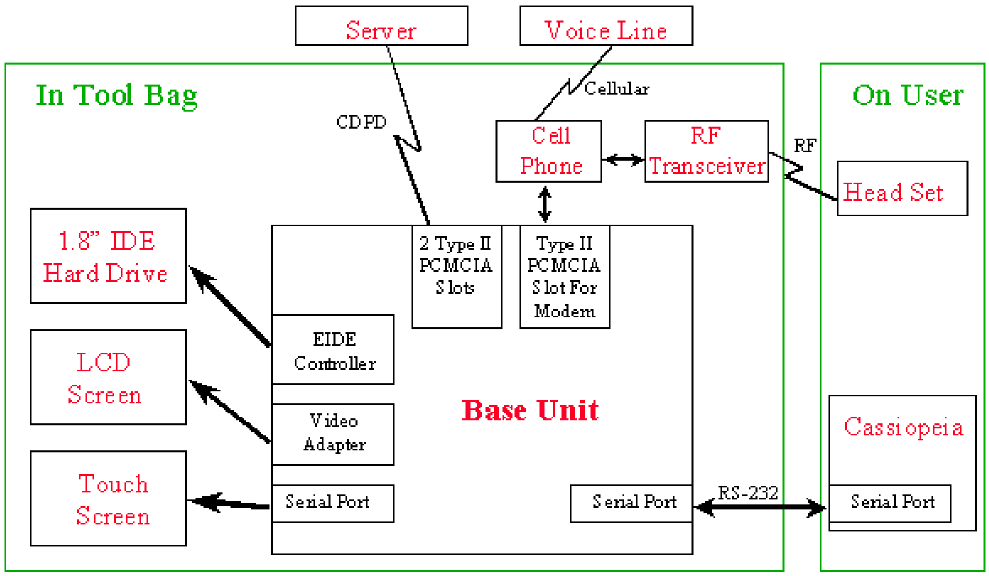


Figure 15. MoCCA Base and Satellite unit Block Diagram

4. User Evaluation

In this section, we will present results of user evaluation studies that were performed.

4.1 VuMan 3

The VuMan 3 user trials were performed at the U.S. Marine base at Camp Pendleton, CA. The user trial results indicated potential savings by reducing maintenance crews from two to one, a decrease of up to 40% in inspection time due to reduced motion of climbing in and out of tight spaces to read / write clipboards, a further 30% of not having to type in hand written notes (Figure 16), and savings of over two orders of magnitude in the weight of maintenance documentation.

4.2 MoCCA

The MoCCA field tests were performed at Digital's facility in Forrest Hills, Pittsburgh, PA. Five FSEs participated in these tests, including performing a set of typical operations related to troubleshooting and repair operations on computing equipment. Each of the FSEs performed all of these operations. The subject systems included printers, motherboards, and networks. The use of MoCCA contributed to a significant saving of time (35 to 40 %), Figure 17. During these field tests, the FSEs used the system for the first time. A larger savings in time is expected with continued use. In addition, MoCCA allowed the FSEs to fix some problems immediately, which otherwise would have required return trips to find and bring back manuals.

4.3 C-130

The C-130 application was evaluated at the 911th Air National Guard, Pittsburgh, PA. The collaboration function of the C-130 application allow for immediate collaboration when problems occur. This results in a significant savings of time compared to a situation without the wearable system.

5. Person Effort

The person effort represents the total amount of effort required to complete all phases of the design methodology. Records were kept throughout the projects to evaluate the design methodology. Table 3 summarizes the design effort among phases in the design. Due to the custom nature of VuMan 3, the design phase required relatively more resources.

Figure 16. Saving Factors From Usage of VuMan 3

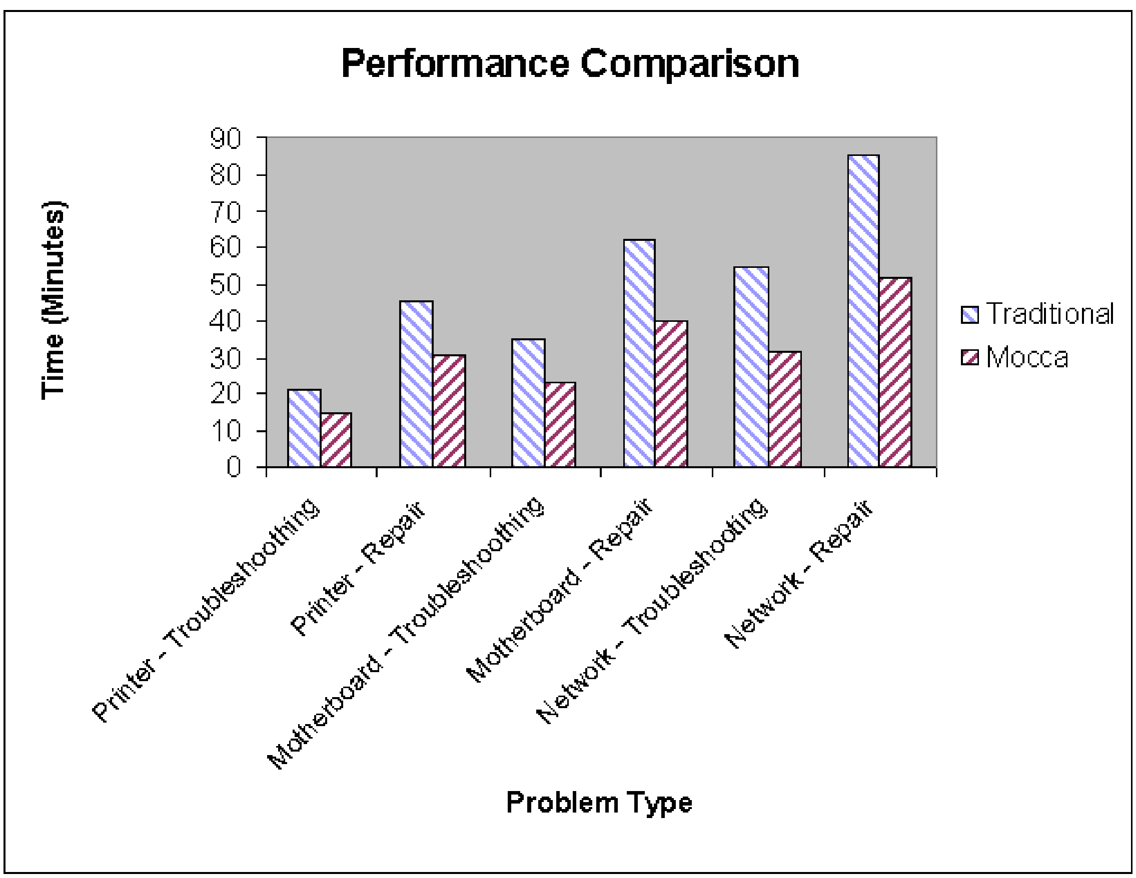


Figure 17. MoCCA Performance Diagram

Artifact	Conceptual Design (%)	Detailed Design (%)	Implementation / Integration / Evaluation (%)	Total Effort (Person Days)
VuMan 3	19	48	33	690
MoCCA	25	39	36	305
C-130	24	38	38	290

Table 3. Design Effort

6. Conclusions

In this paper, we have described a User-Centered Interdisciplinary Concurrent Design Methodology (UICSM), as applied to design and implementation of sixteen novel generations of wearable computers at CMU. The methodology is web - based, defines three phases of a design and implementation cycle, and documentation of the design evolution. We have defined a taxonomy of wearable computers in respect to collaboration application, and presented some evaluation results. While the complexity of the prototype artifacts has increased by over two orders of magnitude, the total design effort has increased by less than a factor of two. Field studies have been conducted and their results reported significant savings. The results of this research should allow us to set the design direction and make appropriate decisions in the future design of advanced wearable computer systems.

7. Acknowledgements

We would like to acknowledge the support provided by DARPA (DABT 63-95-C-0026), NSF Mobile Computing grant, and the Institute for Complex Engineered Systems at Carnegie Mellon University.

References

[1] A. Smailagic and D. P. Siewiorek, "The CMU Mobile Computers: A New Generation of Computer Systems," Proceedings of the IEEE COMPCON 94, IEEE Computer Society Press, February 1994, pp. 467-473.

[2] D.P. Siewiorek, A. Smailagic, and J.C. Lee, "An Interdisciplinary Concurrent Design Methodology as Applied to the Navigator Wearable Computer System," Journal of Computer and Software Engineering, Vol. 2, No. 2, 1994, pp 259-292.

[3] A. Smailagic, D. P. Siewiorek, et. al., "Benchmarking an Interdisciplinary Concurrent Design Methodology for Electronic / Mechanical Design," Proc. ACM / IEEE Design Automation Conference, San Francisco, CA, 1995.

[4] A. Smailagic, "And Evaluation of Audio - Centric CMU Wearable Computers," ACM Journal on Special Topics in Mobile Networking, Vol. 6, No. 4, 1998.

[5] Smailagic, A., Siewiorek, D.P., Stivoric, J., Martin, R., "Very Rapid Prototyping of Wearable Computers: A Case Study of Custom versus Off-the-Shelf Design Methodologies, " Journal on Design Automation for Embedded Systems, Kluwer Academic Publishers, Vol 3., No. 1, 1998.

[6] M. Billinghurst, S. Weghorst, T. Furness III, "Wearable Computers for Three Dimensional CSCW," Proc. The First International Symposium on Wearable Computers, IEEE Computer Society, Cambridge, MA., October 1997, pp 39-47.

[7] S. Feiner, B. MacIntyre, T. Hollere, A. Webster, "A Touring Machine: Prototyping 3D Mobile Augmented Reality Systems for Exploring the Urban Environment," Proc. The First International Symposium on Wearable Computers, IEEE Computer Society, Cambridge, MA., October 1997, pp 74-83.

[8] A. Smailagic, D. P. Siewiorek, A. Dabura, and L. Bass, "MoCCA: A Mobile Computing and Communication Architecture," Computer Journal, accepted, 1998.