Software Development for the Navigator Wearable Computers

D. Siewiorek, A. Smailagic

EDRC 18-47-94

Proceedings Portable By Design Conference, Santa Clara CA, Penton Publishing, February 1994.

SOFTWARE DEVELOPMENT FOR THE NAVIGATOR WEARABLE COMPUTERS

Daniel P. Siewiorek Asim Smailagic

Engineering Design Research Center
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

The paper describes experiences in the design and implementation of a wearable computer, Navigator, focusing on software development. Navigator is a wearable computer with head mounted display, speech recognition, wireless telecommunication, and global position sensing. Navigator includes five off-the- shelf boards, three custom boards, and five customized interconnection interfaces. The progression of subsystem development across five software/hardware platforms, each successively closer approximations to the final platform, is described. Cross development platforms allowed software development to proceed before the final hardware was available. The paper concludes with overall design lessons learned and directions for future implementation and research.[1]

1 Introduction

New paradigms such as the spread sheet allowed the user to interact with their data on an item-by-item basi* looking for patterns and applying "what if scenario. Wearable computers deal in information rather than programs, serving as tools in the user's environment much like a pencil or a reference book. Just as personal computers allow accountant and bookkeepers to merge their information space with their workspace (i.e., a sheet of paper), wearable computers allow mobile processing and the superimposition of information on the user's workspace. Sensors make the wearable computer an active part of the environment. Information can be automatically accumulated by the system as the user interacts with and modifies the environment, thereby eliminating the costly and error-prone process of information acquisition. Carnegie Mellon University has designed and manufactured three generations of wearable computers: VuMan 1 [1], VuMan 2 [2], and Navigator [3]. The next two generations, VuMan 3 and Navigator 2, are under development. Mobile computers are characterized by a modular architecture, which can be custom configured to a particular application.

2 Architecture of Wearable Computers

Wearable computers tightly integrate telecommunications, sensors, speech/gesturing/displays for the human/ computer interface, real time software, and low power electronics housed in a conformable/light-weight package. A wearable computer consists of a number of modular, interconnected components, customized for the task being performed. Example *modules* include:

- Display, such as the head mounted Private Eye.
- Speech input (microphone) and Speech recognition, for interpreting user commands.
- Position sensing, which determine the user's location.
- Motion sensing, to recognize actions being performed by the user.
- Wireless telecommunication, for transmission of data to/from remote databases or computational servers.

Also, there may be a number of stationary devices for providing infrastructure, such as:

- Position sensing, for detecting the location of the user within a geographic area.
- Centralized databases, to maintain up-to-date information.

3 The VuMan Wearable Computers: Series of Embedded Systems

Wearable computers are dedicated to a variety of applications. Therefore, one can not expect that one software model is appropriate for every application. However it is likely that there one model will work well for a class of applications. The software system adopted for the first two generations of wearable computers was a custom-designed information system. Their primary goal was to provide referential information. Table 1 shows the characteristics and attributes of the CMU wearable computers.

Over a 12-week period during the summer of 1991 four

This research has been supported by the Engineering Design Research Center, an NSF Engineering Research Center at Carnegie Mellon University, and the Advanced Research Project Agency.

designers representing electronic, mechanical, industrial design, and software disciplines conceived, designed, and manufactured 30 copies of VuMan 1 in a total of 229 person days of effort. VuMan 1, Figure la, was designed and implemented to be a small wearable computer weighing less than one kilogram for displaying construction blueprints. VuMan 1 allows the user to maneuver (scroll up/down, left/ right; zoom in/out) through a set of blueprints using a simple, multi-button interface and a head mounted display, Reflection Technology's Private Eye [4].

VuMan 2, developed during the fall of 1992, required about half the effort but yielded a factor of four improvement in cost, power consumption, weight and functionality. Composed of only five chips, VuMan 2 provides a menu driven user interface for displaying a variety of information including maps, images, and text. VuMan 2 can display maintenance information, Figure lb, or, with the change of a PCMCIA card, allow a user to locate a person or a building on campus using limited information such as the person's names, the building's name, or general directions.

VuMan 2 is composed of *embedded software,* that includes: test code, screen display, screen scroll, low-level graphical support, simulator, campus/building map translators, on-board embedded monitor, user interface for campus navigation, and a smaller maintenance manual application. VuMan 1 software consisted of only the first three functions.

Currently a ruggedized version of VuMan 2, incorporating new housing designed to withstand shock, temperature, water, and dirt is under development. VuMan 2R will be programmed to support technical inspection checklists, trouble shooting logic charts, and maintenance procedures. Also, under development is VuMan 3, which will extend capabilities to include a higher performance processor (386SL-20), cache memory, hardware power management, and two PCMCIA slots. In addition to a Rash EPROM, another PCMCIA device can be supported in a modular manner such as a spread spectrum radio, a disk drive, etc. Table 1 shows the characteristics and attributes of the CMU wearable computers.

During the spring of 1993, 20 designers produced Navigator with 50% more effort than VuMan 1 but with over an order of magnitude more functionality.

4 The Navigator Wearable Computer

A modular "mix-and-match" architecture allows multiple configurations for Navigator, Figure lc. The display is driven by a 25 MHz Intel 386 processor running X Windows [5] on top of the Mach operating system [6]. Composed of a 16 MB main memory and a 85 MB disk, this processor also runs the Sphinx 1 speech recognition system [7]. A cellular phone provides wireless communications and a Global Position Sensing module provides position information. The applications user interface code is written in Tel [8], and integrated with Sphinx 1 and GPS code, both written in C. An example of the Navigator screen showing the CMU Campus and a path between two locations is shown on Figure 2.

Mach was selected as the primary multi-tasking operating system, in part due to ease of extension and availability of local expertise. Sphinx 1 was also created at CMU. Other speech recognition alternatives focused on end applications and did not have user definable vocabularies. A common graphical/user interface environment based on a customized X server was implemented, because X is a standard with local expertise. Once the decision to use an existing display manager was made, there was really only one choice: the X Window system. While other systems exist for UNIX-derived operating systems, none is more used or portable. In order to provide the broadest flexibility to the application programmers, a X server conforming to the MIT X Consortium's standard has been provided. We believe X allowed us to develop applications quickly, rather than designing custom software [9].

A subset of the interdependent design constraints for Navigator is depicted in Figure 3. The *software constraints* include:

- AT bus for OS
- X Windows restricts OS options
- Tel requires X Windows
- User interface with speech input requires an A/D board

The first three constraints above significantly contributed to increasing the overall housing dimensions, due to the resources that the Mach operating system and X Windows required.

5 Concurrent Development Platforms

The Navigator software evolved over five platforms, Figure 4:

- Workstations, for initial design and algorithm verification
- PC, for development and verification of functionality in target operating system
- Dual PC system, an interim evaluation system to determine the performance enhancement potential of a dual processor solution.
- Open air target system, comprised of the final system's electronic components interconnected on a desktop, to verify operation on the final hardware

Table 1: Attributes of CMU Wearable Computers

Artifact	VuManl	VuMan2	VuMan2R	VuMan3	Navigator 1
Delivery Date	Aug91	Jan 93	Mar 94	May 94	Jul93
Number of Units	30	6	8	8	3
Processor Speed	8 MHz	13 MHz	13 MHz	20 MHz	25 MHz
Memory	0.5 MB	1.25 MB	4.5 MB	40 MB	85 MB
Input	3 button	3 button	dial	dial	speech/ mouse
Display Resolution	720x280	720x280	720x280	720x280	720x280
Communications Phone	-	-	Serial Port	Radio	Cellular
Lines of new code	1800	4700	3100	TBD	38000
Dimensions (in)	2.5x5.5x12	1.5x4x4.4	1.5x4x4.4	2x4x5.8	10x7x2.75
Complexity	24 chips	5 chips	5 chips	8chips	5 boards
Power (W)	3.8	1.1	1.2	2.2	9
Battery Life (hours)	4	12	12	7	8
Weight (lbs)	3.3	0.5	0.5	0.7	9

platform

- Final wearable computer system

The initial software development was performed on RISC-based workstations running the Mach operating system due to their availability to the design team. The initial version of Sphinx 1 was ported to the SPARC and MIPS platforms. The MIPS platform provided excellent profiling analysis tools. Two problems had to be addressed: partitioning the software to maximize parallelism while minimizing communications, and deciding on a target platform. A detailed analysis of the code was performed to find modules along which the software could be partitioned. These analysis revealed three major modules in the Sphinx 1 software: Computation of Cepstrum Coefficients, Vector Quantization, and Search. The relative times spent in each of these modules as well as the communication requirements between partitions was analyzed using the Pixie facility [10].

Since the initial architecture of the Navigator system was comprised of two 80386 processors, one running the DOS operating system and the other running Mach, two 80386 PCs were configured with DOS and Mach. The software was ported to this platform from the workstations and development continued using the PCs.

The next platform was an open air system comprised of the final hardware components augmented by an Ethernet interface, console monitor, and keyboard. It allowed software integration prior to availability of the final packaged wearable system. Therefore, cross development platforms allowed software development to proceed in parallel with the hardware and housing development.

6 Concurrent Design and Task Dependencies

Throughout the design and development of the Navigator

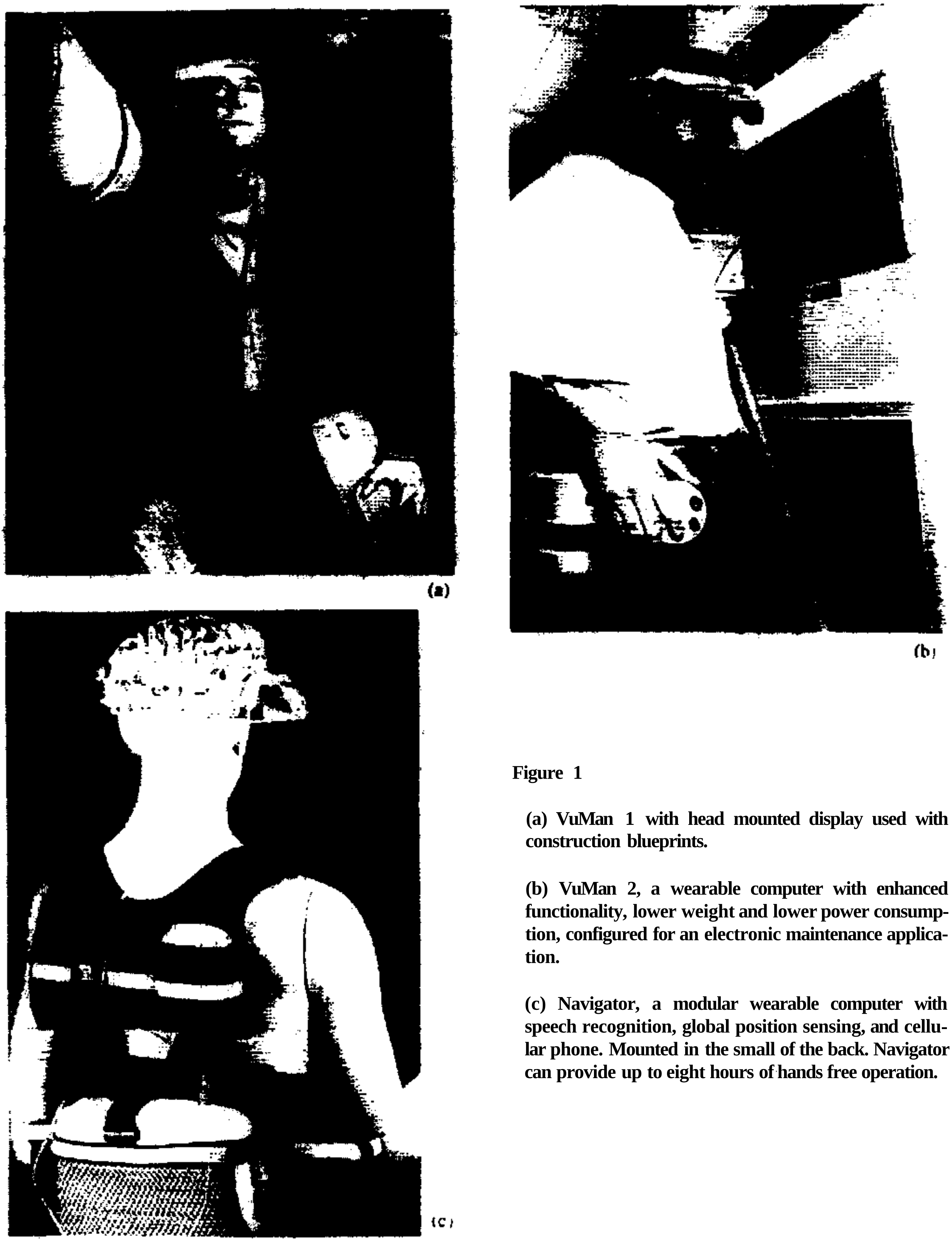
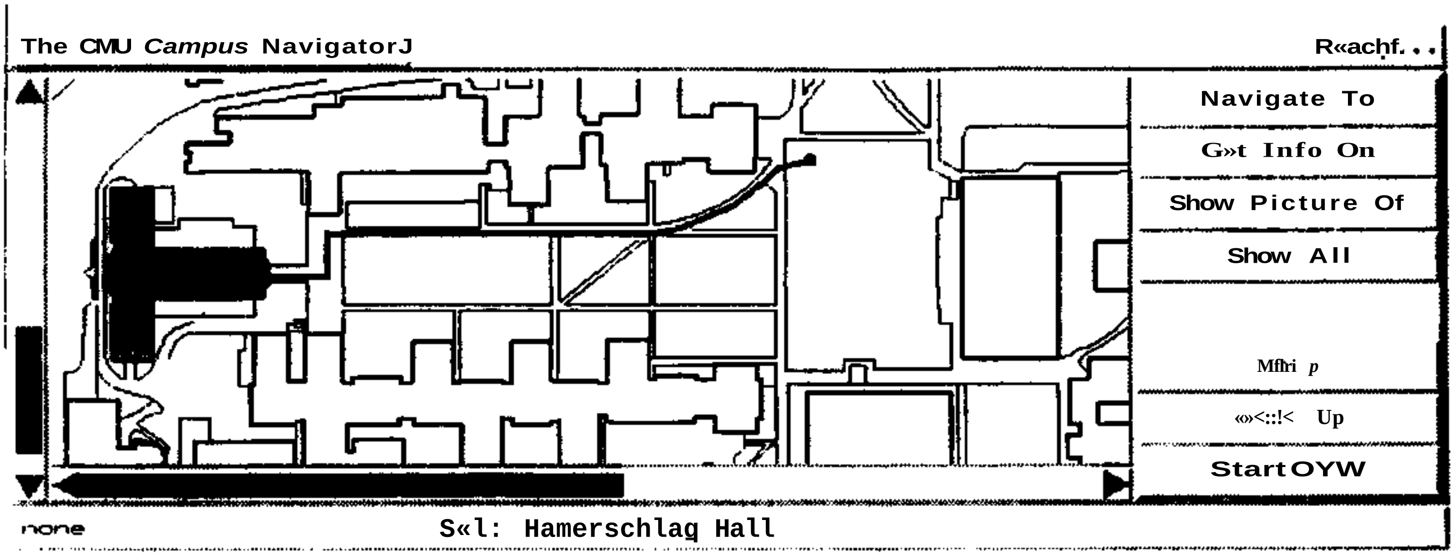


Figure 1

(a) VuMan 1 with head mounted display used with construction blueprints.

(b) VuMan 2, a wearable computer with enhanced functionality, lower weight and lower power consumption, configured for an electronic maintenance application.

(c) Navigator, a modular wearable computer with speech recognition, global position sensing, and cellular phone. Mounted in the small of the back. Navigator can provide up to eight hours of hands free operation.



S«l: Hamerschlag Hall

Figure 2 Navigator user screen, with the CMU Campus map and a path generated from the user to Hamerschlag Hall

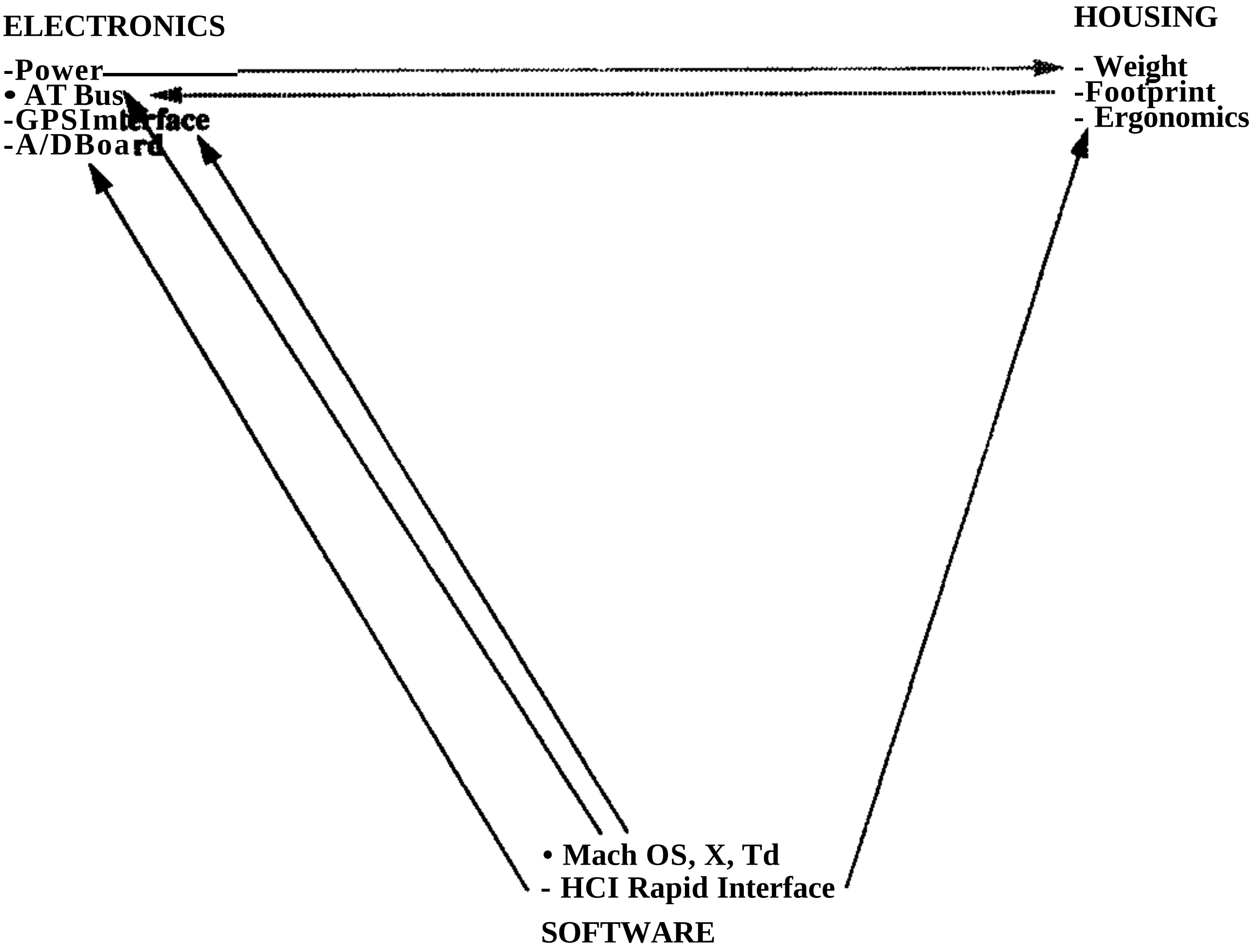


Figure 3 A chart of design constraints

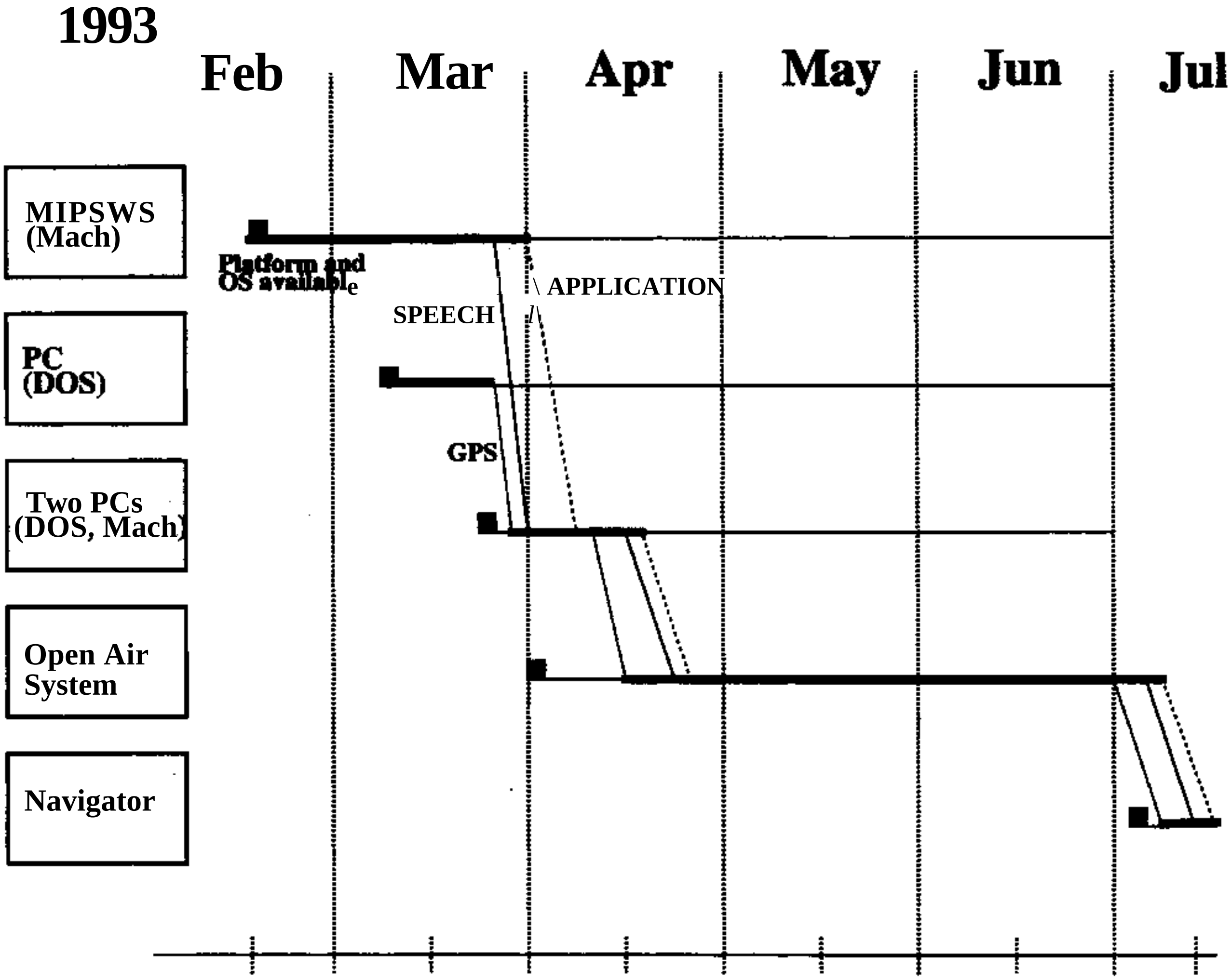


Figure 4:The Subsystems development progression through five platforms in the Navigator project

wearable computer an Interdisciplinary Concurrent Design Methodology (ICDM) [3] is followed. The goal of the methodology is to allow as much concurrency as possible during the design process. Concurrency is sought in both time and resources. The entire design cycle is divided into phases. Activities within a phase proceed in parallel, but are synchronized at phase boundaries. Figure 5 illustrates how the design process proceeded from the initial, Technology Survey, to the final, System Integration phase, and also identifies task dependencies, interfaces between groups, and potential critical paths that might require more resources.

7 Summary and Conclusions

In this paper experiences in the design and implementation of the latest generation of CMU wearable computers, Navigator, is presented and its software development process described. We believe the phased development process provides useful guidance to those interested in undertaking similar projects in mobile/portable computing. The implemented wearable computers and their evaluation study help demonstrate the viability of our approach.

The major lessons learned include:

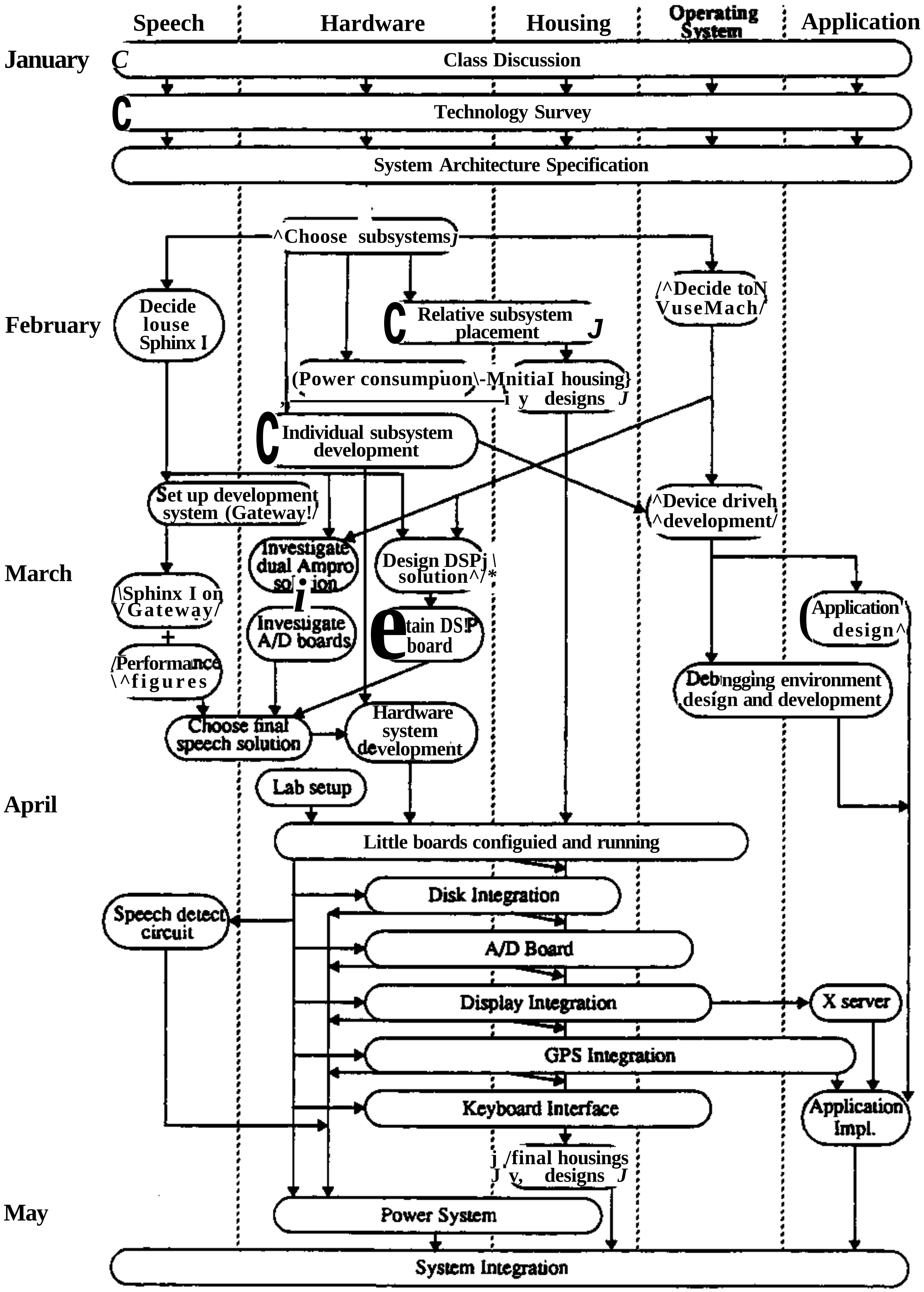


Figure 5 Task dependency graph

- More prototyping is needed before choosing the user interface system
    - Tel may have been a fortunate choice
- Should stress test design decisions before implementation
    - creating/destroying windows vs. raising/lowering windows
- Acquire more end-user input
    - due to the human factor in wearable computers, these systems must be rapidly prototyped and evaluated by actual use
- The development platforms contributed significantly to the concurrent design
- The Mach operating system had more capability than required, and more effort is needed to get a lite operating system for mobile computing

Our current activities should advance the state of mobile computing further:

- The PEE [11] instrumented application code to account for where time is being spent, and obtaining procedure call trees
- Hard disk usage analyzed and unnecessary files removal
- Static power consumption of major hardware subsystems (CPU board, A/D board, hard disk Private Eye display, and power supply) measurement
- Mach kernel modification to take advantage of the hard disk standby mode

Initial exploration of specifications for Navigator 2 has commenced. A 66 MHz Intel 486 class microprocessor should have sufficient performance provide real-time speech recognition. New, more capable head mounted display (from Virtual Vision) will be used, and feedback from our users will provide further specifications.

Mobile computer systems of the future are going to be highly customized. A modular architecture can be custom configured to a particular application, as demonstrated by both the VuMan and Navigator classes of mobile computers.

8 Acknowledgments

We thank Janaki Akella, Chris Castleback, Allen Dutoit, and Prashant Rao for their contributions to the VuMan 1 project. For their contributions to the VuMan 2 project, we thank Drew Anderson, Chris Aycock, Minesh Desai, Roy Maxion, Jay Nigen, Rahul Parekh, Grant Reed, and John Stivoric. The contributors to the Navigator project included: Jim Beck, Forrest Chamberlain, Jason Lee, Tom Martin, Brian Noble, John Stivoric, David VanRyzin, Ali Reza Adi Tabatabai, and Jim Zelenka.

9 References

[I] J. Akella, A. Dutoit, and D.P Siewiorek, "Concurrent Engineering: A Prototyping Case Study," *Proc. Third IEEE Int'l Workshop Rapid System Prototyping*, IEEE Computer Society Press, Los Alamitos, CA, 1992.

[2] A. Smailagic and D.P. Siewiorek, "A Case Study in Embedded System Design: The VuMan 2 Wearable Computer," *IEEE Design and Test of Computers*, Vol. 10, No. 3, pp. 56-67, September 1993.

[3] D.P. Siewiorek, A. Smailagic, J.C.Y. Lee, and A.R.A. Tabatabai, "An Interdisciplinary Concurrent Design Methodology as Applied to the Navigator Wearable Computer System," *Journal of Computer and Software Engineering*, Vol. 2, No. 2, 1994.

[4] A. Becker, "High Resolution Virtual Displays," *Proc. SPIE$_t$* Vol. 1664, Society of Photooptical Instrumentation Engineers, Bellingham, WA, 1992.

[5] MTT X Consortium, *X Window System, Version II, Release 5*, **MIT** Laboratory of Computer Science, Cambridge, MA, 1990.

[6] R. Rashid et al., "Mach: A System Software Kernel," *Proc. COMPCON Spring '89*, IEEE Computer Society Press, Los Alamitos, CA, 1989.

[7] K.F. Li, H.W. Hon, MJ. Hwang, R. Reddy, "The Sphinx Speech Recognition System," *Proc. IEEE ICASSP*, Glasgow, UK, May 1989.

[8] J. Ousterhout, "A/z *Introduction to Tel and* 7fc," Addison-Wesley, NY, 1994.

[9] C.V. Kantarjiev et al., "Experiences with X in a Wireless Environment," *Proc. ofthe Usenix Symposium on Mobile and Location-Independent Computing*, Cambridge, MA, August 1993.

[10] M.D. Smith, *"Tracing with Pixie,"* Reference Manual, Version 1.1, Center for Integrated Systems, Stanford University, 1991.

[II] M. Rao, Z. Segall, "Performance Efficient Parallel Paradigms," *Proc. ofSupercomputing Conference*, 1990.