Matching interface design with user tasks

Modalities of Interaction with CMU Wearable Computers

Asim Smailagic and Daniel P. Siewiorek

To maximize the effectiveness of wearable systems in mobile computing environments, interface design must be carefully matched with user tasks. By constructing mental models of user actions, interface elements may be chosen and tuned to meet the specific software and hardware requirements of specific procedures. This article details specific cases of task study and interface implementation by the Carnegie Mellon University (CMU) wearable computer project over six generations of systems. An abstract model for application interface design is given, with examples of implementation on both embedded systems and general-purpose platforms.

Among the most challenging questions facing mobile system designers is that of human interface design. As computing devices move from the desktop to more mobile environments, many conventions of human interfacing must be reconsidered for their effectiveness. How does the mobile system user supply input while performing tasks that preclude the use of a keyboard? What layout of visual information most effectively describes system state or task-related data? The Engineering Design Research Center at CMU is engaged in research to answer these and similar questions.

CMU has designed and built six generations of wearable computers with a set of generic applications, providing a challenging environment in which to study the impact of user tasks on interface design. Figure 1 and Table 1 summarize the key characteristics and attributes of the CMU wearable computers. System design has been centered around two general methodologies: embedded and general-purpose. The embedded nature of the VuMan series of wearable systems has permitted excellent size and power consumption characteristics without sacrificing flexibility in software interface design. The general-purpose Navigator series, through the use of "off-the-shelf" hardware and software components, offers greater modularity and extensibility while preserving a fully custom-designed user environment. The effectiveness of these systems has been evaluated in real-world contexts for referential and navigational tasks. The VuMan systems have undergone field-testing at the United States Marine Corps maintenance facilities at Camp Pendleton in California. Running a custom hypertext engine, the VuMan serves as a portable document repository, replacing volumes of hardcopy texts. The Navigator has been evaluated for use in the similar setting of aircraft maintenance, where the combination of speech recognition and a VGA heads-up display allows hands-off access to complete aircraft specifications and inspection checklists. Both systems have been tested with navigational applications, in which the wearable computer supplies database information on a local environment and can assist in directing the user from one geographic location to another.

The objective of wearable computer design is to merge the user's information space with his or her work space. The wearable computer should offer seamless integration of information processing tools with the existing work environment. To accomplish this, the wearable system must offer functionality in a natural and unobtrusive manner, allowing the user to dedicate all of his or her attention to the task at hand with no distraction provided by the system itself. Conventional methods of interaction, including the keyboard, mouse, joystick, and monitor, all require some fixed physical relationship between user and device, which can considerably reduce the efficiency of the wearable system.

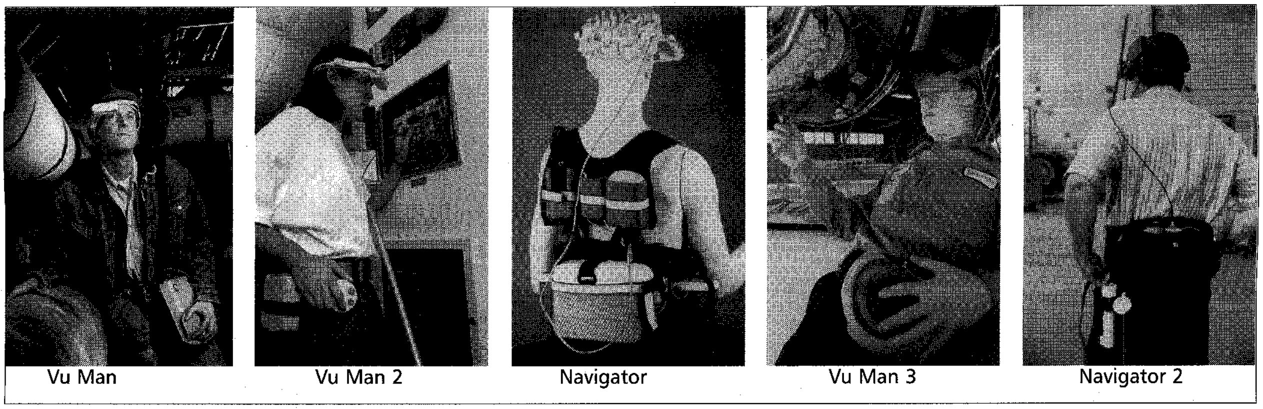
This article describes the factors of human interface in wearable systems design, as addressed by the Engineering Design Research Center wearable computer project. Based on representative examples from these six generations of wearable computers, the article will focus on how the application impacts the design and especially the user interface. First examined is the process of modeling specific user tasks, and establishing a linkage with the system interface that closely mirrors this model. Next is an overview of the VuMan embedded systems, followed by a description of the Navigator general-purpose machines. The article concludes with a summary of information developed through the design and testing of these wearable systems.

Navigation and Checklist Task Models

The efficiency of the human-computer interface is determined by the simplicity and clarity of the mental model suggested by the system. By modeling the actual task as well as the human interface, a linkage can be constructed between user and machine which can be examined to improve

This work has been supported by the Engineering Design Research Center, an NSF Engineering Research Center, and the Advanced Research Project Agency.

1070-9916/96/$05.00 © 1996 IEEE



Artifact Specifications	Vu Man1	Vu Man 2	Navigator 1	Vu Man 3	Navigator 2
Delivery date	Aug. '91	Dec. '92	June '93	Dec. '94	July '95
Number of units	30	6	3	20	8
Embedded/GP	Embedded	Embedded	General-purpose	Embedded	General-purpose
Design style	Semi-custom	Fully custom	Design by composition	Fully custom	Semi-custom
No. custom boards/chip count	1/24	1/5	3/15	2/10	2/29
No. off-the-shelf boards	1	0	5	0	2
Lines of code	1800	4700	38,000	12,000	88,000
Processor	80188-8 MHz	80C188-13 MHz	80386-25 MHz	80386EX-20 MHz	486SX-33 + DSP
RAM/nonvolatile storage	8 KB/512 KB	512 KB/1 MB	16 MB/85MB	1 MB/420MB	12MB/420MB
Input	Three-button	Three-button	Speech/mouse	Rotary w/button	Speech/joystick
Display resolution	720 x 280	720 x 280	720 x 280	720 x 280	640 x 480
Dimensions (in)	10.5 x 5.25 x 3	4.75 x 4.5 x 1.37	7.25 x 10x3	5 x 6.25 x 2	5.8 x 10.7 x 3.2
Power (W)	3.8	1.1	7.5	2	7.5
Weight (lbs)	3.3	0.5	9	1.75	4
Magnitude of design activity	Innovative	Innovative	Innovative	Innovative	Innovative
(a) Number of designers	4	6	21	16	23
(b) Number of CAD tools	16	16	7	7	8
(c) Person-months of effort	12	6	28	23	33
Quantitiy of artifacts fabricated	30	6	3	20	8
(a) Electronic part count	45	12	8 boards	2 boards	2 boards
(b) Housing fabrication	Vacuum-forming	SLA/molding	Pressure-forming	Molding/machining	Molding/machining

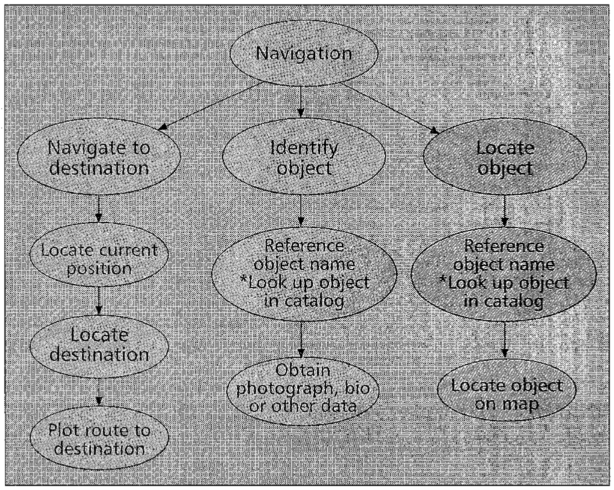
■ **Figure 1.** *The evolution of EDRC wearable computers.*

the overall efficiency of the wearable system. We begin with the assertion that for wearable systems to be efficient, the mental model of the interface design must closely parallel that of the user task; there must be minimal interference or obstruction posed by the computer in completing jobs.

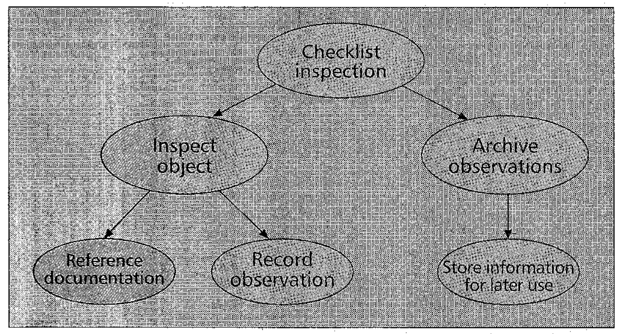
Application development for the CMU wearable computers has focused on tasks that can currently be performed without the use of a computer and fall into the broad categories of navigation and checklist inspection. In preparing the wearable system applications, the design objective was to consolidate the sometimes large data sets required to perform these tasks into a conveniently accessible reference medium. While the information storage devices used on the VuMan and Navigator serve to reduce the physical size of the reference material, this is not the only improvement sought in the design of the systems.

Navigation Tasks

The concept of navigation as used when discussing a site or local area may be divided into three basic operations: traveling to a destination, identifying objects, and finding the location of objects (Fig. 2). The first of these captures the traditional process of identifying, on a map, a current location, a destination, and a path between the two. Depending on the topography of



■ **Figure 2.** *General model of the navigation task.*



■ **Figure 3.** *General model of the checklist task.*

the site, familiarity with the area and other factors, this task may represent varying degrees of difficulty. Typically, one would need some means of reference to establish the current position; traditionally, this involves landmark comparison using a map. The subject would then have to establish the position of the destination. Last, an efficient path would need to be computed between the two. This last problem presents a significant challenge, because certain characteristics of site topography may seriously impact the nature of an efficient path.

One deficiency in the above interpretation of navigation is the assumption that one knows what it is one is seeking, and would recognize it upon arrival. A subject placed on a site with which he or she lacks familiarity might wish to obtain information about general items of interest, such as dining or communications facilities; one might also seek more specific data, such as office information for an individual. This type of referencing activity may require large data sets, traditionally contained in books or immobile database kiosks. Also, under many conventional methods of reference, only textual information on the object of the search is available; the process of converting building names or office locations to a recognizable object remains a task for the subject.

This more comprehensive view of navigation requires the subject to gather several distinct pieces of information from independent sources and manually compile them. The outwardly straightforward process of navigation contains several subtasks, potentially difficult for the subject unfamiliar with the surroundings, which cannot be consolidated using conventional means.

Checklist Inspection Tasks

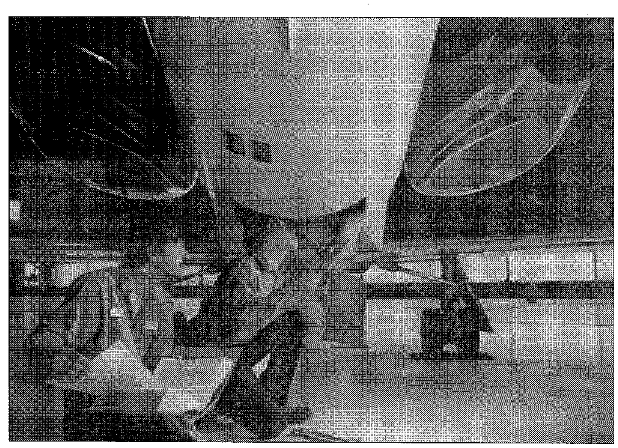
The checklist inspection includes a collection of tasks including documentation reference, inspection and annotation, and archival storage of observations (Fig. 3). The first item, document reference, may require formidable volumes of data for complex systems. The large array of data items necessary for a comprehensive vehicle inspection might preclude any attempt at consolidation of sources; images of an air intake might reside in a manual separate from relevant part numbers, which in turn cannot be found in the maintenance checklist manual, and so forth. For time-critical operations, this process of cross-referencing sources introduces nontrivial losses of efficiency (Fig. 4).

The document referencing task occurs in parallel with an organized system of inspection. In this process, the subject follows a checklist of instructions, confirming the status of, or supplying information on, each item in the list. This may require a simple visual observation or a more complex examination, possibly including the use of diagnostic tools; the subject may require access to the above-mentioned reference materials at any time during this inspection. After gathering information on the item to be inspected, the subject must record the results of the analysis for later use. This procedure may also require the presence of cumbersome inspection books or other recording items not well suited to compact work areas or environmental factors such as rain. In some cases, the combination of these reference and recording materials can require the help of additional personnel to manage the bulky manuals.

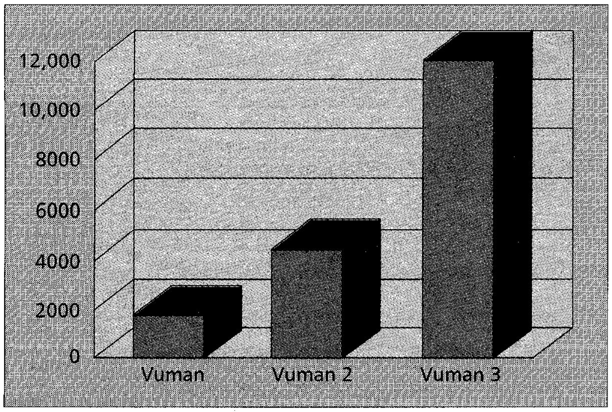
At the completion of an inspection, the information gathered by the subject must typically be stored for future reference. In some record-keeping applications, this process requires manual transcription of the recorded observations into logbooks, typewritten reports, or computer databases. Duplicating the contents of the checklist in this manner introduces a nontrivial amount of overhead to the total inspection time, and presents a new opportunity for error in the transcription process.

Modeling the User Interface

The wearable system interface should act to complement a user's experience while performing a job. By examining the correspondence between the mental model of a task as perceived by both the user and system designer, progress should be made towards reducing inefficiencies caused by the introduction of the computing device. Observable factors which reveal inefficiency at the interface level



■ **Figure 4.** *Technician consulting bound documentation during inspection.*



■ **Figure 5.** *Lines of code for embedded system applications.*

include hesitation [2], which can occur when task structures as expected by the user do not correspond to the structure as perceived by the designer. Such model mismatches can be avoided through consultation with prospective users at design time, so as to avoid embedding a misinterpretation of the task model in the interface.

In an evolutionary design cycle, such as that employed with the VuMan HyperText System (VHTS) package developed over two generations of VuMan systems, continued examination of the mental model correspondence emphasizes usability over feature set upgrades. According to Nielsen [3], such iterative interface design methods, practiced over at least three versions of a package, should act to improve the usability of a system substantially. VHTS3.0, released in January 1995, implements a series of design enhancements requested by representatives of the intended user population at Camp Pendleton. Further improvements scheduled for VHTS3.1 include on-screen feedback on battery life, progress indicators, and upgrades to the data transfer system.

Hypermedia, or hypertext, has been demonstrated as an alternative to more traditional media for reference purposes [4]. The ability to integrate graphical and textual data along with dynamic document navigation suits hypertext to tasks such as the reference system required for checklist inspections. Hypertext models in which users can store and manipulate data using a form interface have already been introduced; these systems are typically found on desktop workstations [3]. VHTS captures these features using a hypertext engine which displays documents written in a portable scripting language. The VuMan HyperText Language (VHTL) specification includes support for graphics, form, fields and interdocument hyperlinks.

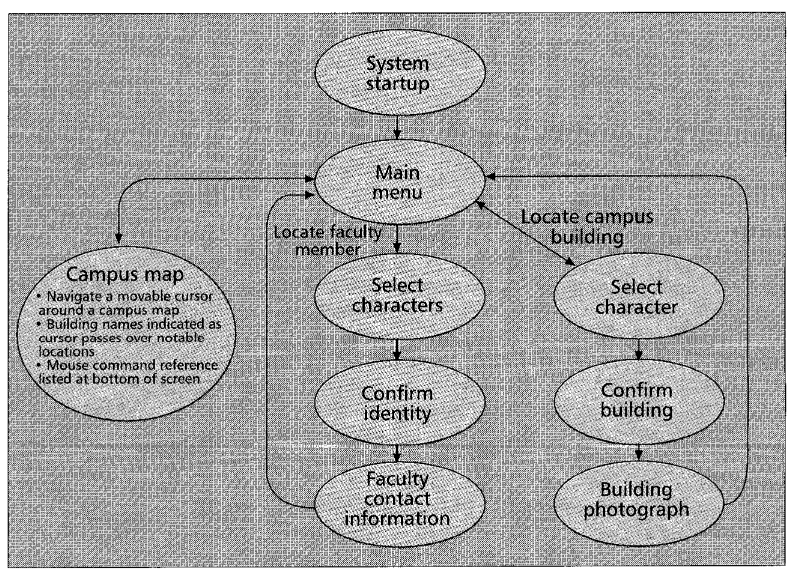
To study the correlation between task models and user interface in checklist referential systems, we turn to the VuMan HyperText System. As discussed earlier, checklist inspections may be considered to consist of document reference, observation recording, and archival subtasks. As noted above, a significant degree of interaction between the reference and recording tasks may occur as an inspector consults documentation during a procedure. Such documentation may not be limited to textual data, so graphical support must be included in this referential system. Last, the structure of the inspection system must be flexible to allow multiple paths through the procedure.

The VuMan Embedded Systems

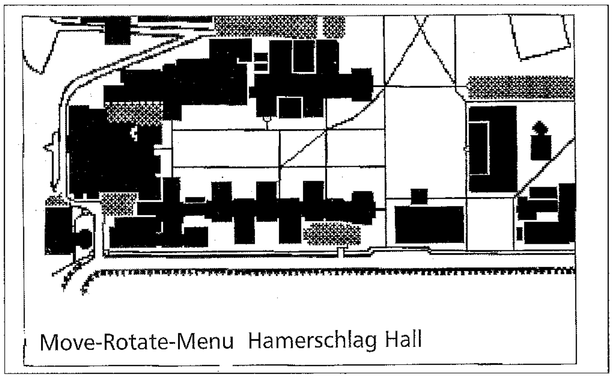
The target application areas for the VuMan series of wearable computers consist largely of referential or checklist systems. In this model, the wearable system serves as a repository of mobile information, capable of recording structured observations. Of particular interest are two software packages which capture this functionality: the VuMan campus tour, and the VuMan HyperText System.

In 1992, with the delivery of the VuMan 2 wearable computer [5], an embedded application was developed to assist users in navigation about the Carnegie Mellon campus. The primary features of this system (Fig. 6) include a navigable overhead map of the campus (Fig. 7), a database to locate faculty members (Fig. 8), and a utility to display images of buildings based on name (Fig. 9). Because the VuMan 2 was custom-engineered, the choice of interface remained entirely at the discretion of the development team. The selected method included a three-button mouse tethered to the main CPU by a serial line; the controls on the mouse correspond to screen-dependent instructions, the names of which are displayed for reference in the lower left corner of the screen (Figs. 7, 10, and 11).

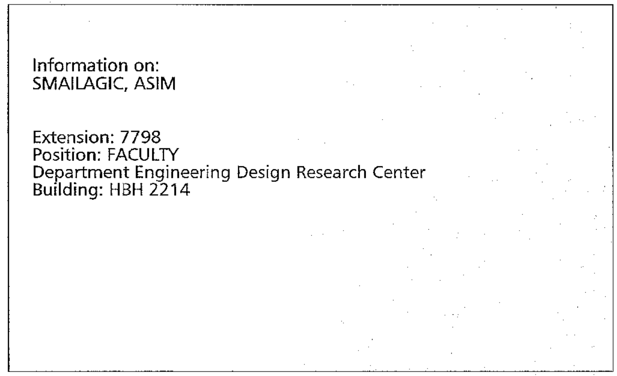
The campus map portion of the tour consists of an autoscrolling bitmap display of the campus, with a movable arrowhead cursor controlled by the user (Fig. 7). While viewing this map, the mouse controls allowed directional rotation of the cursor, forward movement of the cursor, and an exit from the map. As the cursor passes over each item of interest on the map, the name of the object appears at the bottom of the screen. The faculty member database and the building locator both employ the same indexing interface; the user identifies the first one or two characters (depending on whether the search is for a building or faculty member) in the name of the target of their search. The interface to this index is one-dimensional in nature; the user selects characters in a linear alphabetic array, using the mouse buttons to traverse the length of the menu (Fig. 10). Once the search has been narrowed using this criterion, the user selects from a list of possible matches, and is presented with a screen displaying contact information (for faculty) or a scanned photograph (for buildings). This indexing method results in an organizational hier-



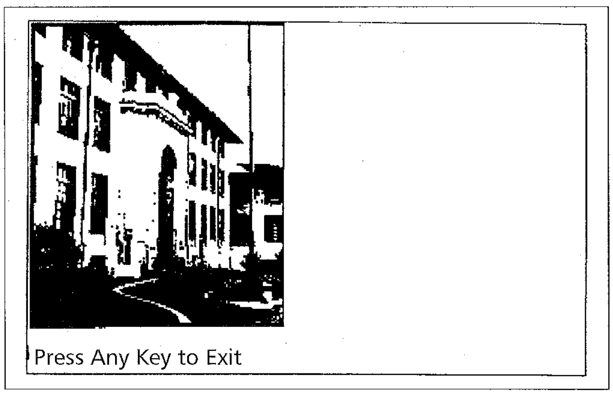
■ **Figure 6.** *Finite-state representation of VuMan 2 campus tour application.*



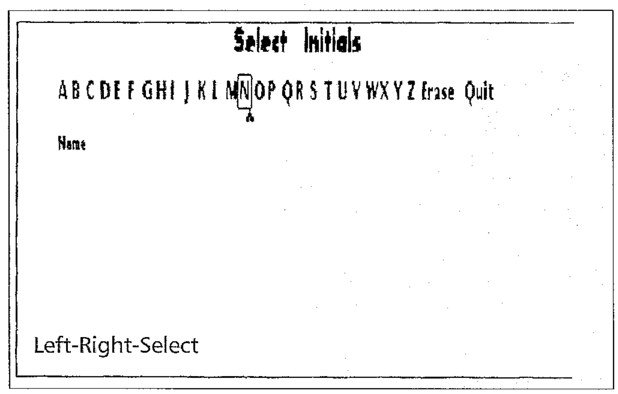
■ **Figure 7.** *Campus Tour map.*



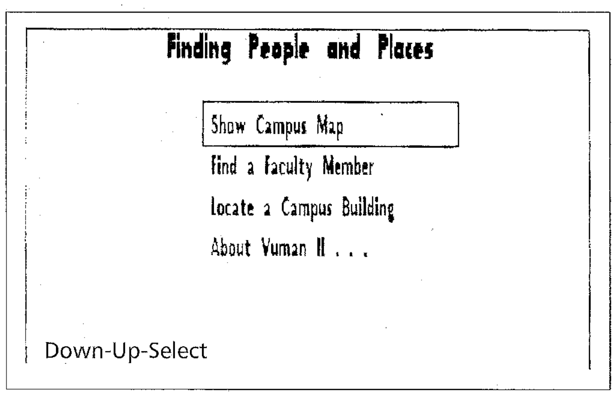
■ **Figure 8.** *Campus Tour faculty information.*



■ **Figure 9.** *Campus Tour building info.*



■ **Figure 10.** *Campus Tour character index.*

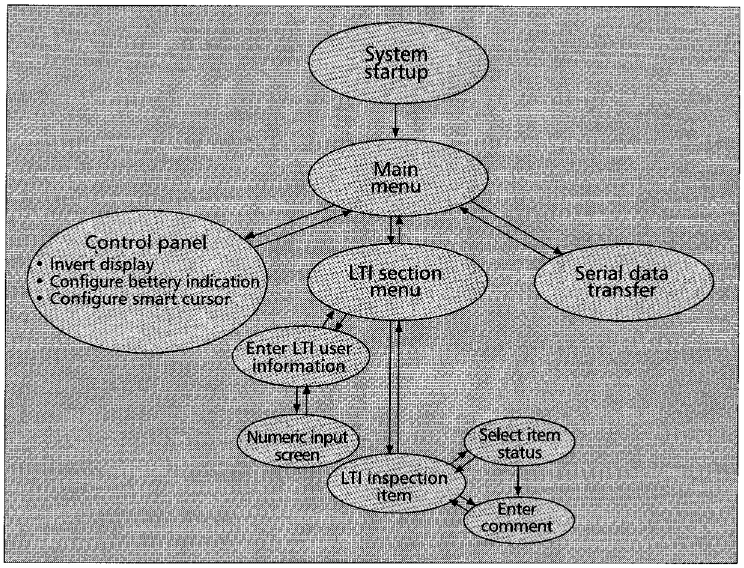


■ **Figure 11.** *Campus Tour main menu.*

archy of information, which allows the user to access what can be considered a multidimensional data array. The first, most general, dimension, in which the user specifies the first character in a faculty or building name, must be traversed in a linear fashion, with order n performance (where n is the displacement from the middle of the alphabet of the selected character). In the case of searching for a faculty member, a second degree of character indexing follows the first, also with order n performance. Finally, the last degree of indexing occurs with the selection of a faculty or building name from a linear list of matches to the search parameters.

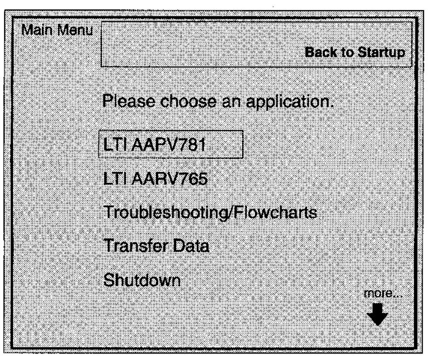
Following the development of the campus tour, a flexible implementation system was conceived, VHTS. Modeled on existing hypermedia implementations, VHTS allows dynamic document navigation with the addition of forms-based user input similar to standards such as HyperCard and the HyperText Markup Language. VHTS represents a significant advance in modularity for the VuMan, as the creation of new document systems involves only the authoring of new text files written in VHTL. One document system deployed for field use by the United States Marine Corps, the Limited Technical Inspection (LTI), implements a checklist of over 600 items. This application represents an order of magnitude increase in screen content over the Campus Tour, while requiring less physical storage space due to the text-intensive nature of the LTI, which can be stored as character data formatted at runtime, as opposed to full-screen bitmaps. An additional advantage of the VHTS page storage system is the ability to construct virtual pages, or pages whose size exceeds the physical dimensions of the display. The VHTS Viewer decomposes these pages into smaller physical pages at runtime, providing a scrolling mechanism to allow browsing of the entire page.

The LTI application showcases many VHTS features, including flexible document navigation, smooth interaction with lower-level electronics, and built-in data transfer support. As the user begins the LTI (Fig. 12), a Main Menu of high-level options (Fig. 13) allows navigation to a Control Panel, one of two inspection procedures, information about the VuMan, or data transfer options. Choosing the Control Panel (Fig. 14) jumps to a page of selectable options allowing flexible control over the display environment; users may select the orientation of the display, configure battery status indicators, or set the status of several other interface features. By proceeding to either of the LTI sections, the user begins an organized walk-through inventory of an amphibious vehicle. The user begins by entering status information about the inspection procedure itself, including the serial number of the technician performing the inspection; this information can be input in a numeric entry screen in which the user selects numerals from a list of digits. The user then moves directly to the inspection items, categorized into 12 groups. Each screen within a section (Fig. 15) may contain one or two inspection items, the status of which may be noted in a form-like manner. The user selects the status field for the item under consideration, and chooses from a menu of options, including Serviceable, Unserviceable, On ERO (Extended Repair Order), or Missing. At this point, the hypertext nature of VHTS becomes apparent; a selection of Unserviceable jumps directly to a menu of comments describing the nature of the problem; other choices simply cause the viewer to jump to the next item scheduled in the inspection. To simplify operations, standard interface conventions are used for each screen shown in Fig. 16. Items at the top allow sequencing between logical portions of the inspection or returning to the Main Menu. Items along the right-hand side of the screen allow sequencing through multiple physical screens comprising one logical portion of the inspection. All items on a screen are in a circular list that is accessed by an oversized dial (Fig. 17) that can be rotated clockwise or counterclockwise, coupled with a selector switch which surrounds the dial. VuMan 3's mechanical controls are an intuitive interface to the linear list that can be operated in any orientation, with the user wearing gloves, and even through the cloth of a coverall pocket containing the VuMan. Upon completion of the LTI, the collected data must be stored for record-keeping. The user can, at the completion of an inspection, connect the VuMan via serial cable to a DOS PC and upload the LTI inspection items. Extraction utilities on the PC can isolate specific instances of a keyword, which can be useful in the generation of inspection reports.

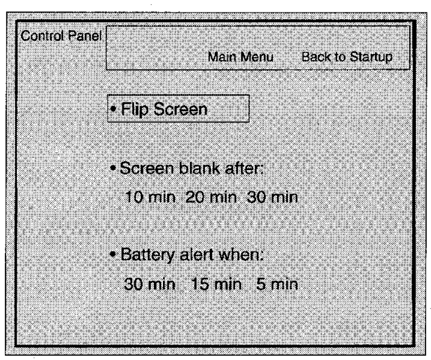


■ **Figure 12.** *Finite-state representation of VuMan 3 LTI application.*

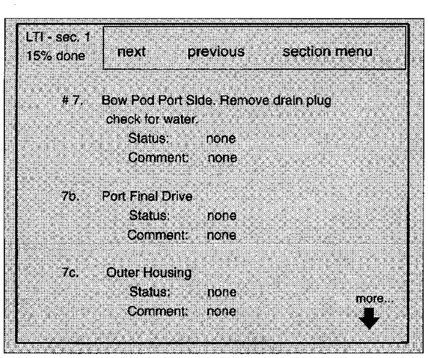
At the completion of a preliminary trial of the VuMan 2 running a VHTS LTI for the Marines at Camp Pendleton, several design changes were proposed to improve the usability and durability of the VuMan design. VuMan 3, the product of these improvements, features a ruggedized casing, on-board power supplies, a rotary dial for primary input, and a docking station which allows simple charging and data transfer through a new serial port. These advances in human interface design greatly enhanced the effectiveness of the VuMan system for field work similar to the Marine LTI. Despite the large number of inspection items present in the LTI, the simple procedure of selecting an on-screen status and a comment from a list of options, if necessary, reduces the typical inspection time from roughly between four and six hours to less than three.



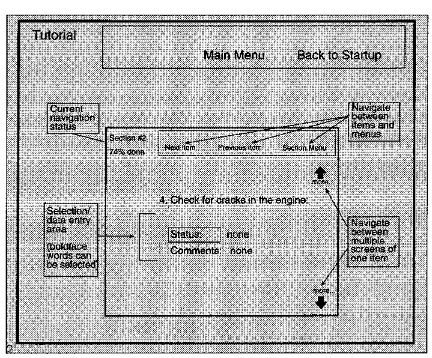
■ **Figure 13.** *LTI inspection main menu.*



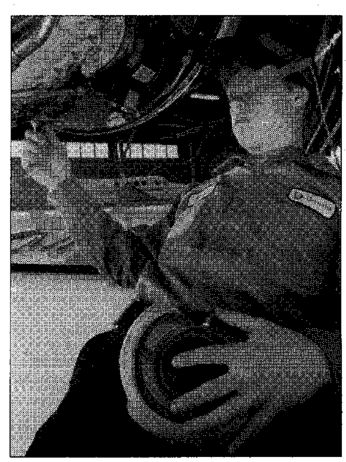
■ **Figure 14.** *LTI control panel.*



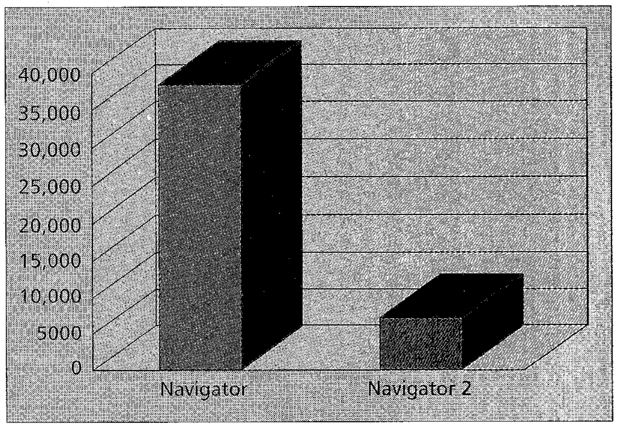
■ **Figure 15.** *LTI inspection screen.*



■ **Figure 16.** *LTI tutorial screen*



■ **Figure 17.** *VuMan 3 dial and selectors.*



■ **Figure 18.** *Lines of code for general-purpose system applications.*

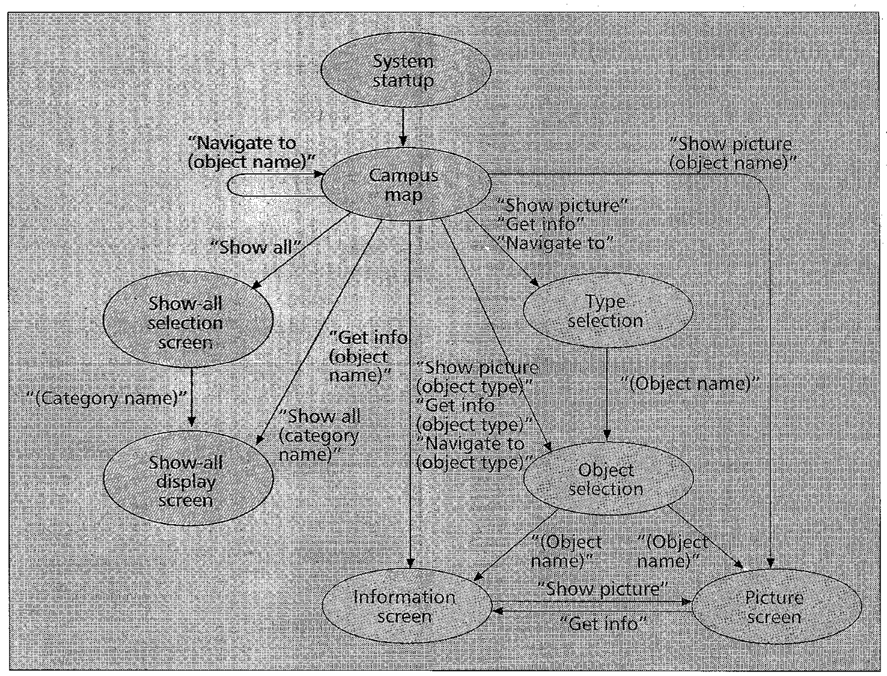
The Navigator General Purpose Systems

Although embedded systems offer an extremely flexible platform for hardware- and software-level interface design, the absence of a conventional operating system precludes the use of many existing human interface tools. Packages such as Tcl/Tk, which can speed development times and provide a consistent application appearance for graphical interfaces, traditionally operate under established environments such as the X Window System, and are unusable in an embedded design. Navigator [6], a general-purpose system running Mach on an i386 processor and using X11 as the graphical imaging system, allowed for the development of a new Campus Tour using off-the-shelf interface design components.

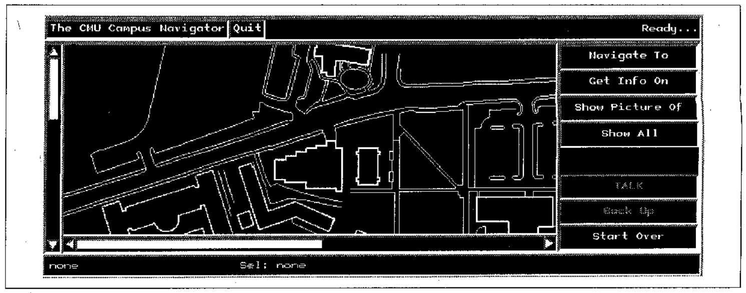
At startup, Navigator brings up an X server under the Mach kernel and the Campus Tour (Fig. 19 [1]) begins, employing both the SPHINX Speech Recognition System [7] and a portable mouse for input. The user identifies their current location on a map of the Carnegie Mellon campus (Fig. 20) using the mouse (this location is automatically marked when the global positioning system — GPS — is employed). The user then issues one of several instructions to Navigator using spoken instructions or a series of button selections with the mouse (Fig. 21). For example, the user might wish to visit a particular building on campus; the spoken command "navigate to Hamerschlag Hall" serves as well as clicking the Navigate To button (Fig. 20), then indicating Building as the destination type (Fig. 21), then selecting Hamburg Hall as the destination name. Both methods invoke an on-screen plot of the most efficient path from the present location to the destination (Fig. 22). The use of speech allows users to jump over the intervening screens, thereby saving time. While the mouse must sequence through each screen, the mouse mode is used to train the user in the vocabulary. The training avoids user frustration when trying to guess the legal sentences supported by the grammar in the speech recognition system. Once the vocabulary is learned, the user can use the more efficient speech recognition mode. Other options include the ability to search for information about a faculty member, which can produce a photograph of the selected individual (Fig. 23), or requests to locate nearby items of interest. Users can, for instance, request the locations of nearby rest facilities, libraries, or food services (Fig. 21). The extensible nature of the Navigator system allows for the use of a GPS receiver to provide automatic location marking on the Campus Map.

Building on some of the concepts used to construct the Navigator, the design of a new general-purpose system directed towards multimedia applications was undertaken. The Navigator 2 uses an i486 Epson processor card to run a voice-controlled custom aircraft inspection application (Fig. 24). The speech recognition system, with a secondary manually controlled cursor, offers complete control over the application in a hands-free manner, allowing the operator to perform an inspection with minimal interference from the wearable system.

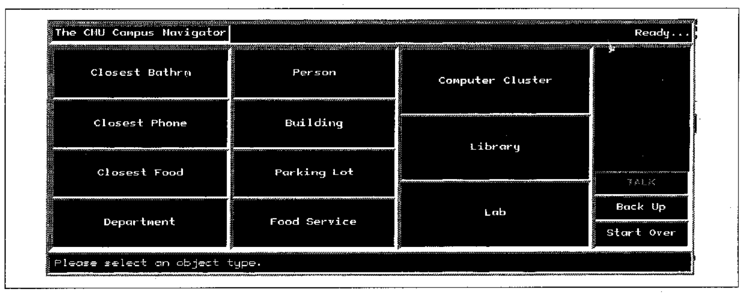
At startup (Fig. 25), the application prompts the user for their choice of activating the speech recognition system or not. The user then proceeds to the Main Menu. From this location, several options are available, including online documentation, assistance, and the inspection task (Fig. 26). Once the user chooses to begin an inspection, information about the inspection is entered (Fig. 27), an aircraft type to examine is selected, and the field of interest is narrowed from major features (Left Wing, Right Tail — Fig. 28) to more specific details (individual panes in the cockpit window glass — Fig. 29). The area covered by each defect as well as the type of defect, using a How Malfunctioned code such as Corroded, Cracked, or Missing, are recorded. To maximize usability, each item or control may be selected simply by speaking its name or, in the case or more complicated phonemes, a desig-



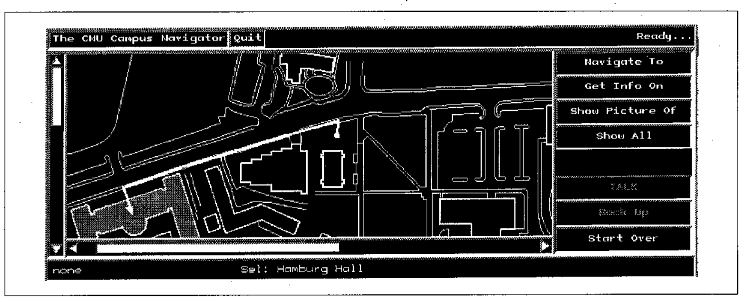
■ **Figure 19.** *Finite-state representation of Navigator Campus Tour application.*



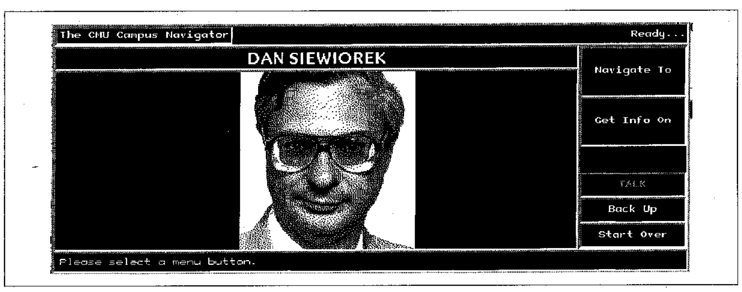
■ **Figure 20.** *Navigator campus map.*



■ **Figure 21.** *Navigator object selection.*



■ **Figure 22.** *Navigator campus map showing path to destination.*



■ **Figure 23.** *Navigator faculty info.*

nating numeral (the recognition engine interprets three more reliably than KC-135). This two-dimensional selection method, in which defect locations are specified on a planar region, differs from the input system used in the LTI. On the VuMan, the linear selection method used to augment the rotary dial is better suited to dealing with discrete items or statuses. As with VHTS-based systems, the Navigator 2 can upload recorded data to a remote receiver. Using one of the four Personal Computer Memory Card Industry Association (PCMCIA) expansion slots, a wireless Ethernet transceiver may serve as the data connection between the two machines.

User Interface Assessment

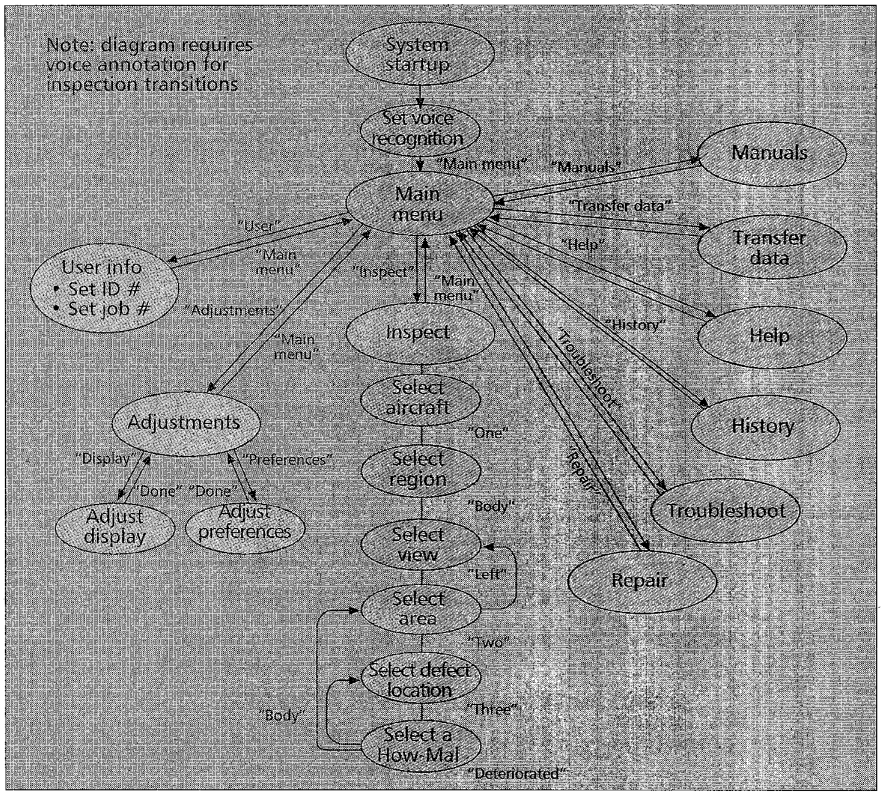
Although the number of quantifiable metrics suited for interface evaluation is small, a series of basic observations provide a means for comparison. One characteristic of an application interface is the number of user actions required to perform a given subtask. We define a subtask as an operation, possibly consisting of multiple inputs, which a user completes in the process of performing a larger coherent task. For example, in the course of performing an inspection, a user might wish to return from their present location within an application to the main menu. This subtask may require a single input (perhaps a voice command or an onscreen button) or multiple inputs (backing out through a hierarchy of categories to reach the top, or main, level). We assert that an application requiring few inputs will allow a user to dedicate more attention to the job at hand, while a larger number of inputs will require more concentration on the computing system. A comparison of equivalent inspection subtasks in the VuMan 3 LTI and Navigator 2 applications appears in Fig. 30 and Table 3. Note that speech input is not itemized separately, as touchpad cursor control and voice control perform identical functions on the Navigator 2 application.

One interesting note concerning the Navigator 2 application is the number of actions required to perform some actions associated with locating and setting the status of inspection items. The larger number of actions required by the Navigator application as compared with the VuMan LTI may be attributed to the larger number of hierarchical levels used to organize physical regions. This choice of categorization serves to break down the large surface area of an airplane into more manageable sections, but increases the cost of navigating between regions of the inspection.

A similar comparison of navigation applications, such as those available on the VuMan 2 and Navigator, is presented in Fig. 31 and Table 4. Included is a separate listing for speech input in the Navigator Campus Tour; the SPHINX recognition engine accepts complex commands which allow some subtasks requiring a series of manual inputs to be executed with a single phrase.

The contrast between manual and speech input methods in the Navigator application suggests that the availability of complex commands through the SPHINX system may significantly improve usability by reducing the required number of actions per subtask. A cost of this feature in Navigator is severe performance deterioration, approaching a response rate of 8 x input duration, due to the slow 25 MHz i386 processor. As an example, a typical complex command may require two to three seconds to enunciate; this results in a 16–24 s delay before execution of the spoken command. Added to this burden is the overhead introduced by the Tcl interface toolkit used for the graphical display. For these reasons, we must factor in the subjective aspect of system latency in our evaluation of usability.

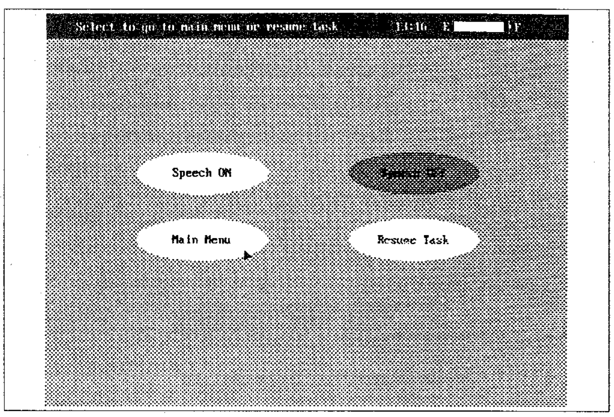
By means of comparison, the Navigator 2 TERI discrete word recognition system running on a 33 MHz i486 responds in less than 800 ms. This improvement in system response rate comes at the cost of the aforementioned complex instructions provided by the SPHINX engine. Spoken input accepted by



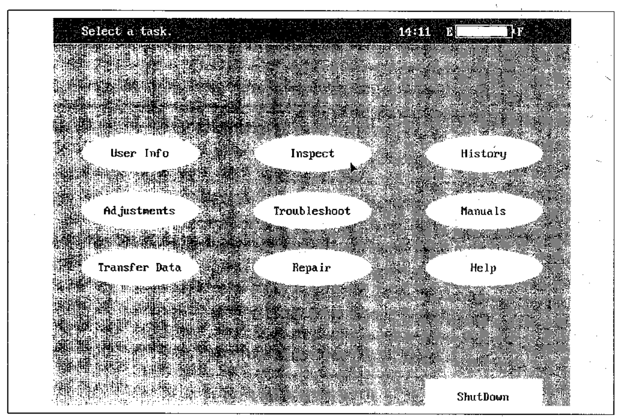
■ **Figure 24.** *Finite-state representation of Navigator 2 Boeing maintenance application.*

the Navigator 2 application is restricted to single words, thus precluding the use of complex commands as is available with the Navigator campus tour. A direct consequence of this characteristic is that, for purposes of counting the number of inputs required to perform a task, the use of speech with the Navigator 2 requires the same number of user actions as does manual or mouse input. For applications such as the checklist inspection, however, this trade-off may not be too costly, because the model of the task suggests that user actions are performed most often as discrete operations rather than in batches. The ability to process complex instructions could, for the set of operations dealing with moving from one region of the inspection to another, measurably reduce the number of inputs required in certain cases.

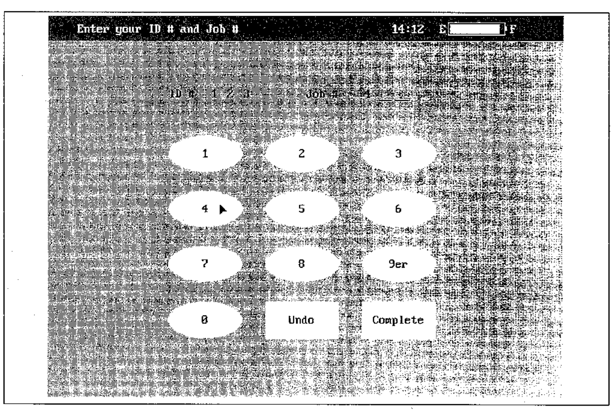
The metric of hesitancy to evaluate interface efficiency [2] can be affected by the number of options between which a user must choose at any given time in the course of



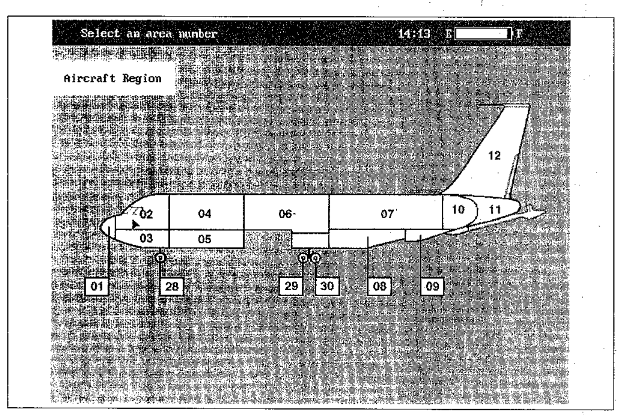
■ **Figure 25.** *Navigator 2 speech selection.*



■ **Figure 26.** *Navigator 2 main menu.*



■ **Figure 27.** *Navigator 2 user info.*



■ **Figure 28.** *Navigator 2 region selection.*

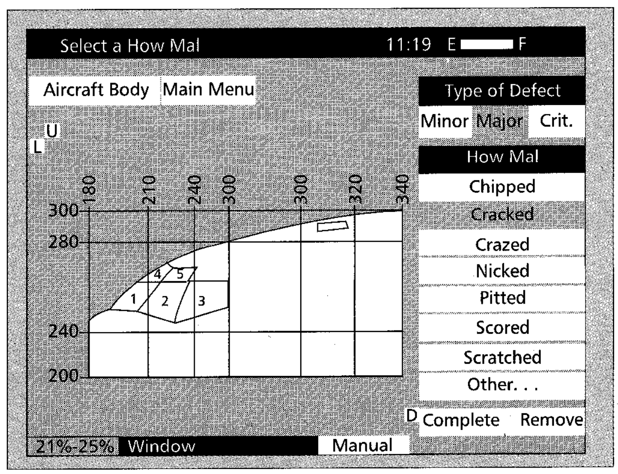
Attribute	VuMan	VuMan 2	VuMan 3	Navigator	Navigator 2
Application	Map display	Campus Tour	LTI checklist	Campus Tour	Boeing maintenance
Application type	Navigation	Navigation	Checklist	Navigation	Checklist
Development languages	80188 Assembler, C	80188 Assembler, C	80386 Assembler, C	C, Tcl	C++
Lines of code	1800	4700	12,000	30,000	8000
Required storage	180 KB	384 KB	448 KB	4 MB	2 MB
Storage type	ROM	PCMCIA flash RAM	PCMCIA flash RAM	IDE hard drive	PCMCIA hard drive
Graphics tools	Custom	Custom	Custom	Tcl/Tk, Motif	Microsoft GUI Toolkit
Operating enviroment	Embedded	Embedded	Embedded	Mach 3.0	MS-DOS
Input	Buttons	Mouse	Dial/buttons	Speech/mouse	Speech/pressure pad
Text entry	None	Character menu	Menu	Speech	Speech, onscreen keypad
Output display	Private Eye HMD	Private Eye HMD	Private Eye HMD	Private Eye HMD	Virtual Vision VGA HMD
Communications support	None	Serial	Serial	Cellular	Spread-specrum radio

■ **Table 2.** *Wearable computer software attributes.*

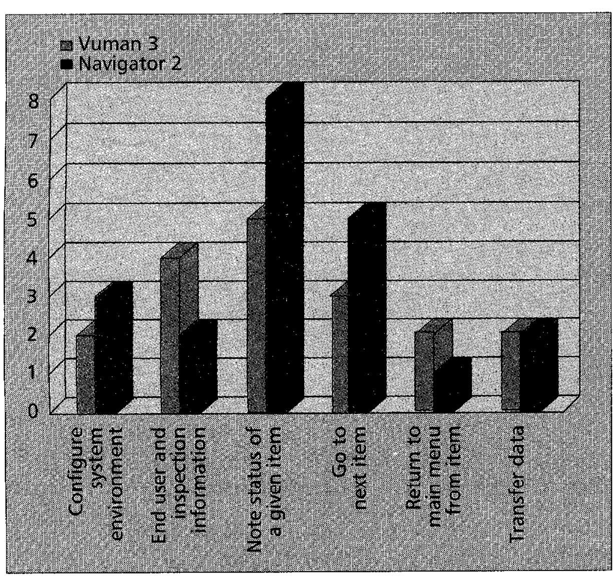
operating the system. The number of commands available per screen serves as an adequate measure of this characteristic; the results of these measurements appear in Table 4. This information can be enhanced by considering the presence of repeated or global commands; options that appear in consistent locations with consistent functionality throughout the application should reduce the amount of time required for the user to examine the available options on each screen. The number of global commands for each application appear in Figure 31.

Conclusion

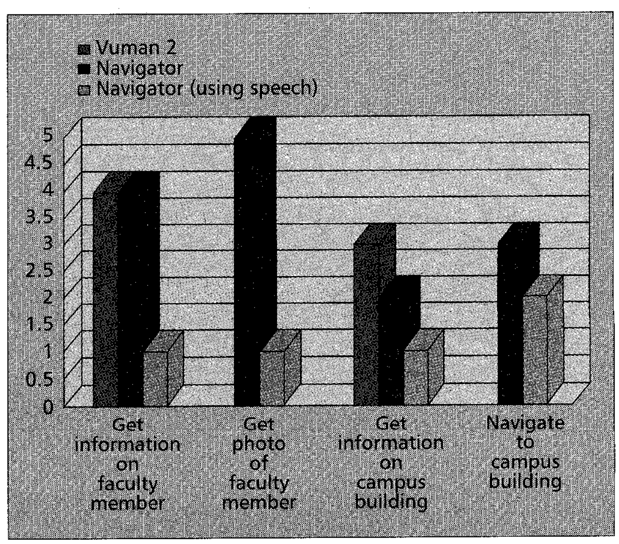
In this article we study how the application impacts the design and especially the user interface, and present cases of task study and interface implementation. Generic application modes of interaction are presented and evaluation



■ **Figure 29.** *Navigator 2 inspection screen.*



■ **Figure 30.** *User actions required to perform inspection subtasks (assuming minimum linear searching).*



■ **Figure 31.** *User actions required to perform navigation subtasks.*

Task	Vuman 3	Navigator 2
Configure system environment	1. Choose "System Options" 2. Select setting	1. Choose "Adjustments" 2. Choose "Adjust Display" or "Adjust Preferences" 3. Select setting
Enter user and inspection info	1. Choose LTI 2. Choose "Inspection Info" 3. Choose item to set 4. Enter numeric information	1. Select "User Info" 2. Enter ID # and job #
Note status of a given item	1. Choose LTI 2. Choose section 3. Locate item (linear with position in section 4. Choose "Status" 5. Set status	1. Choose "Inspect" 2. Choose aircaraft type 3. Choose region 4. Choose view 5. Choose area 6. Choose defect location 7. Choose type of defect 8. Choose "How Mal"
Go to next item	*If within section:* 1. Linear with location in section *If not within section:* 1. Choose "Section Menu" 2. Choose section 3. Locate item (linear with position in section)	*If within area:* 1. Select item on defect location screen *If not within area:* 1. Select "Aircraft Body" 2. Select area 3. Select item If not within region: 1. Select "Aircraft Body" 2. Select "Aircraft Region" 3. Select region 4. Select area 5. Select item
Return to main menu from item	1. Choose "Section Menu" 2. Choose "Main Menu"	1. Choose "Main Menu"
Transfer data	1. Choose "Transfer Data" 2. Select data set to be transferred	1. Choose "Transfer Data" 2. Choose "Transfer Data"

■ **Table 3.** *User actions required to perform comparable subtasks in cheklist inspection applications.*

Task	Vuman 2	Navigator	Navigator (using speech)
Get info on faculty member	1. Choose "Find a Faculty Member" 2. Select first character of last name 3. Select second character of last name 4. Choose name from list of matches	1. Choose "Get info on" 2. choose "Faculty Member" 3. (optional) Enter indexing characters of last name 4. Choose name from list of matches	1. Say "Get info on <Faculty Name>"
Get photo of faculty member	N/A	1. Choose "Get Info on" 2. Choose "Faculty Member" 3. (optional) Enter indexing characters of last name 4. Choose name from list of matches 5. Choose "Show Picture"	1. Say "Show picture of <Faculty Name> "
Get Info on campus buliding	1. Choose "Locate Campus Building" 2. Select first character of building name 3. Choose building name from list of matches	1. Select building on campus map 2. Choose "Get Info on"	1. Say "Get info on <buliding name>"
Navigate to campus building	N/A	1. Mark present location on campus map (if not using GPS) 2. Choose destination building on map 3. Choose "Navigate to"	1. Mark present location on campus map (if not using GPS) 2. Say "Navigate to <builiding name>"

■ **Table 4.** *User actions required to perform comparable subtasks in navigation application, including data for speech interface.*

measures provided, user interaction with the application is characterized, and mental model coordination between the user and the application task is introduced. Examples of implementations on both embedded systems and general-purpose platforms are offered.

Based on the results of user trials that we have performed for heavy military vehicle maintenance and aircraft inspection, we refine our user interface models and set up appropriate benchmarking procedures. We are planning to introduce additional modalities of interaction, such as gesture recognition, and other forms of displaying information, and to provide support for collaboration.

We should point out the fundamental nature of this research, and its importance will increase as mobile computing becomes more pervasive. The six generations of wearable computers are evolutionary steps in the quest for new ways to improve and augment the integration of information in the mobile environment.

The fundamental nature of this research, and its importance will increase as mobile computing becomes more pervasive. The six generations of wearable computers are evolutionary steps in the quest for new ways to improve and augment the integration of information in the mobile environment.

Acknowledgments

For their contributions to the design of user interfaces for wearable computers, we thank Len Bass, Jeremiah Blatz, Chris Kasabach, Dewitt Latimer, John Stivoric, and Gautam Vallabha.

References

[1] F. Chamberlain, "Secondary Storage Options for a Wearable Computer," Masters Rep., Carnegie Mellon Univ. Dept. of Electrical and Comp. Eng., 1994.

[2] R. Maxion and A. deChambeau, "Dependability at the User Interface," *Proc. 25th Int'l. Symp. on Fault-Tolerant Computing*, June 27–30, 1995, Pasadena, CA, pp. 528–35.

[3] J. Nielsen, "Iterative User-Interface Design," *Computer*, Nov. 1993.

[4] T. Landauer *et al.*, "Enhancing the Usability of Text Through Computer Delivery and Formative Evaluation: The SuperBook Project," A. Dillon and C. McKnight, Eds., *Hypertext: A Psychological Perspective*, 1995.

[5] 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, Sept. 1993, pp. 56-67.

[6] D. P. Siewiorek *et al.*, "An Interdisciplinary Concurrent Design Methodology as Applied to the Navigator Wearable Computer System," *J. Computer and Software Eng.*, vol. 2, no. 3, Nov. 1994, pp. 259–92.

[7] K. F. Li *et al.*, "The Sphinx Speech Recognition System," *Proc. IEEE ICASSP*, Glasgow, U.K., May 1989.

Biographies

ASIM SMAILAGIC is a faculty member at Carnegie Mellon University's Engineering Design Research Center. He has participated in serveral major projects that represent milestones in the evolution of computer system architectures, from Carnegie Mellon's Cm* Multiprocessor System and Edinburgh Multi-Microprocessor Assembly (EMMA) in the '70s to Carnegie Mellon's current project on wearable computer systems. His research interests include rapid prototyping of wearable computers, mobile computing, hardware-software codesign, parallel computing, audiovisual interfaces to computer systems, and concurrent design methodology. He is a member of the ACM, the IEEE Computer Society, and the IEEE.

DANIEL P. SIEWOREK [F] is a professor in the School of Computer Science and the CIT Department of Electrical and Computer Engineering at Carnegie Mellon University. He received the B.S.degree in electrical engineering (summa cum laude) from the University of Michigan, Ann Arbor, in 1968, and the M.S. and Ph.D. degrees in electrical engineering (minor in computer science) from Stanford University, Stanford, California, in 1969 and 1972, respectively. His current research interests include computer architecture, reliability modeling, fault-tolerant computing, modular design, and design automation.