Research and Industry: An Overview of Speech Interaction with Wearable Computers in Industrial Applications

Christian Bürgy[1] and James H. Garrett, Jr[2]

[1] Wearable Consult, Unterer Burggarten 17, 69221 Dossenheim, Germany
christian.buergy@wearable-consult.com
http://www.wearable-consult.com

[2] Carnegie Mellon University, Department of Civil and Environmental Engineering
5000 Forbes Avenue, Pittsburgh, PA, 15213-3890, USA
garrett@cmu.edu
http://www.ce.cmu.edu/~garrett

Abstract. In recent years, we developed several speech-controlled wearable computer systems from commercial off-the-shelf products. Using this approach, we could focus on the application at hand and on how to best integrate commercial speech products into customized software. In this article, we summarize some of our experience, introduce an interaction model that allows for the transfer of system engineering experience between different applications and domains, and finally illustrate some of the feedback that we got from industry while working with them on applying wearable computers to industrial applications.

1 Introduction

As wearable computers emerge to become useful tools, and speech recognition and speech synthesis get more and more reliable, industry recognizes the applicability of speech-controlled wearable computers, especially for data collection processes. Speech technology will be amongst the most important interaction means with wearable computers, and it is an interaction modality that enables real hands-free computing, even today.

In this paper, we summarize our experience on speech-controlled wearable computers that we collected in recent years. The first part of this paper discusses the motivation for our research – a formalized design process for these systems based on an interaction model that aids in system design decisions. In the second part of this paper, we describe how speech technology evolved in recent years, and what the demands of industry are, based on feedback we got from research partners and customers.

2 Research: An Interaction Constraints Model

2.1 Research Motivation

Our research group at Carnegie Mellon University [1] has conducted several research projects on speech-controlled wearable computer systems for industrial applications. The projects were from different domains (automotive, manufacturing and civil engineering) and targeted different applications. Table 1 shows the hardware platforms and the software development tools used for these projects. In all of these projects, we chose to only use commercial off-the-shelf (COTS) components wherever we could, and combined those into a functional system. Thus, the main focus was on the conceptualization and implementation of the applications themselves. CISS-VR is an end-of-line support system using speech-controlled virtual reality 3D objects [2]; ICMMS supports progress monitoring on construction sites [3]; mDCT is a support system for mobile landfill monitoring [4], MIA is a system for speech-controlled generation of bridge inspection reports [5]; and SCWC is a speech-controlled mobile support system for vehicle safety inspections [6].

	Wearable Hardware	Development Language	Speech Recognition System
CISS-VR	Xybernaut MA IV	Java	L&H ASR 1600
ICMMS	Xybernaut MA IV	Java	IBM ViaVoice
mDCT	Xybernaut MA IV Magellan GPS	Visual Basic	Dragon Naturally Speaking
MIA	Via II PC	Visual Basic	Dragon Nat. Speaking / IBM ViaVoice;
SCWC	Xybernaut MA IV Proprietary HW	Visual Basic	L&H ASR 1600 / L&H ASR 1500

Table 1. Hardware platforms and software development tools

In these projects, we and the users we intended to support identified speech control as the best and most applicable currently available hands-free interaction modality. In addition, we were able to compare the development efforts for the different components of the wearable computer systems. Table 2 shows the comparison of the development times needed for the different design and implementation tasks of each project and the actual share of development time spent for each part of the system, such as hardware architecture design and implementation, the actual software application, the graphical user interface (GUI), and the speech interaction development. From this comparison, the need for making the speech interaction design process faster and more efficient is obvious: the speech interaction design takes a major share of the development time of the system, but is to a large extent application-independent. As such, the creation of the speech interaction could be formalized and supported similarly to existing GUI design formalizations and guidelines.

Design, implementation and development time	CISS-VR	ICMMS	mDCT	MIA	SCWC
HW architecture	20%	10%	25%	25%	40 %
Application logic & data management	30%	35%	40%	30%	20 %
GUI design for specific device	25%	30%	15%	15%	10 %
Speech interaction design	25%	25%	20%	30%	30 %

Table 2. Design time comparison

Although each project supported specific tasks in different environments and different domains, all five system design processes faced similar challenges in setting up the mobile speech interaction. First, we had to analyze the tasks to be performed, then we determined the appropriate user interaction, and finally, if we selected speech as a possible means of interaction for a specific task, we had to choose the right form of speech interaction with the user. The numbers in Table 2 show that the design of the speech interaction took at least as much time as the GUI design (with the exception of the ICMMS, which included a complex GUI with interactive construction drawings). Speech interaction design also took nearly as much time as the hardware architecture design (either proprietary design or systems design with COTS components), which shows the potential in reducing development times with a formalized process for speech interaction design for mobile systems.

There are guidelines on how to build speech dialogs [7,8], but most of them were set up for an office or lab environment where there is low ambient noise, the user sits in a chair and has his or her full attention devoted to the application (mostly using word processors, spreadsheets, or browsing the web).

Possible solutions for supporting the speech interaction design process would either be a decision support system that suggests forms of interaction to the developer or even an automatic generation tool for the speech interactions. Such systems can take as inputs the specifics about the user, the device being used, the application being performed, the environment in which the application is being used, and the tasks to be performed, and either suggest solution strategies or create application frameworks that have to be implemented and optimized. Similar systems already exist or are in progress for GUI development [9,10]. However, the first step for such a tool has to be a formalized description of these input parameters, especially with respect to the performed tasks, and a common model of speech interaction in this area. Based on this model, we can develop a decision support system that might in the future evolve to such an automatic tool.

2.2 Interaction Constraints Model – Preliminary Assumptions

To approach the problem described in the previous section, we have developed five different system prototypes in five different industrial domains and environments and observed, analyzed, and categorized activities performed by mobile workers using these prototypes. From this experience, we have gained insight about the needs and constraints for speech-controlled mobile IT support with wearable computer systems in industrial applications.

In performing several iterations of system design and systems engineering, we acquired valuable experience and knowledge about these issues – especially on how best to apply speech-control in these industrial environments. In industrial applications, many activities have to be speech-controlled because the potential users employ their hands for their primary task – their actual job. This experience evolved from system to system and was illustrated in the resulting designs, such as the separation of inspection lists into portions that could be managed “at once” (i.e., without scrolling the screen) on small displays; or the use of clearly understandable system state indicators, such as a traffic light that informs the user about the system’s readiness for interaction (see Figure 1). From field tests in real work environments (e.g., shop floors, garages, and construction sites) we have obtained feedback from real users telling us about their needs, likes, and dislikes. With that knowledge, we focussed on building a model of all these tasks to be applied during the design process of wearable computers.

Fig. 1. Examples for small display (12 lines, 20 char.) and GUI design with traffic light symbol

While developing the interaction model, we made the following assumptions about the targeted system designs that a decision tool based on that model should support:

- The decision tool will support the design of speech-controlled wearable computer systems;
- These systems will support workers (blue-collar and white-collar workers) who, at a certain time, do not have a desk at which to work and often need both hands to perform tasks other than operating the system;
- The systems will be used in industrial environments, i.e. noisy, dirty, and rugged;
- The users of these systems typically will not be early adopters of IT devices;
- The use of speech recognition and synthesis and other interaction modalities will be independent of the domain; and
- Since the underlying model will be domain-independent, the activities to be augmented by speech-controlled IT support should be described in a generic form.

Based on these assumptions, we developed an interaction model that could map work situations that are to be supported by speech-controlled wearable computers with situations from other applications and domains. Thus, this model helps in retrieving information on the applicability of specific interaction means based on previously made experience. Underlying the model is the idea to define work situations based on the constraints that occur and their influence on using the wearable computer. As we developed this system, we realized that the whole approach could not only be used for speech interaction but for any interaction with mobile and wearable computer systems, and thus it evolved to the *Interaction Constraints Model*.

2.3 Interaction Constraints Model

The developed *Interaction Constraints Model* [2] maps constraints of specific situations in which mobile IT support is needed to identify user interface components that may be incorporated in the system design. Due to the nature of industrial applications, these situations mostly are work situations, i.e., situations in which users of mobile and wearable computers work at a specific location on the worksite and have to perform an actual job.

This means that the user's interaction with the device is not only constrained by the physical location, but also by the activities that are supported by the device. The importance of location and activity evolved from the opportunity to establish IT support at the actual workplace through wearable computers. The fact that the computer support moved from a central location, such as the desktop or a kiosk-like computer, to "anywhere" on the worksite makes it inevitable during the design process to take into account the location of the mobile worker. The fact that mobile IT support helps to accomplish another activity – the actual job – requires that we view operating a mobile IT support tool only as a secondary task. Thus, this secondary task has to be unobtrusive with respect to the primary task, and must not exhaust the cognitive and physiological capabilities of the worker, such as the available attention that can be given to the device, the number of available hands for operating the device, or just the willingness of the users to use the device while performing another activity.

Interaction Constraints Patterns

Constraint patterns that influence the interaction between the user and the wearable computer system, which we call *Interaction Constraint Patterns* (ICP), can help to describe the characteristics of a specific situation in an application-independent and domain-independent way. In focusing on ICPs, or sets of constraints, and in mapping these constraints to usability information of user interfaces, we can build up a generic description of the conditions of work situations that help to decide on the applicability of specific interfaces for specific situations.

Before computers were mobile, the interaction was mainly influenced by three components: the user, the computing device and the application that was supported by the computer. Now, we face two more categories that have to be added: the environment in which the device is used and the task that the device supports. Thus, the design of mobile IT support is limited by constraints with respect to the kind of *task* to be performed; the *application* for which the task is performed; the influences caused by the *environment* on the execution of the task; the *device* chosen as the supporting hardware platform; and the abilities and work patterns of the *user*.

The level of detail of the constraints collection has a significant impact on the resulting data quality. If the level of detail is too low, i.e., there are too few attributes for each constraint, the data might not be meaningful enough. However, if there are too many details, i.e., there are too many attributes for each constraint, the collected data might not allow designers to find any work situations with the same ICP. For the same reason, we chose limited value ranges for each constraint attribute, i.e., mostly two or three options per value, instead of a broad range of values. Thus, we defined the attributes for the five constraint categories with an ICP as shown in Figure 2.

For the proof-of-concept we analyzed and compared around 300 work situations of 15 projects on mobile and wearable computers and derived this formalization for the *Interaction Constraints Patterns*.

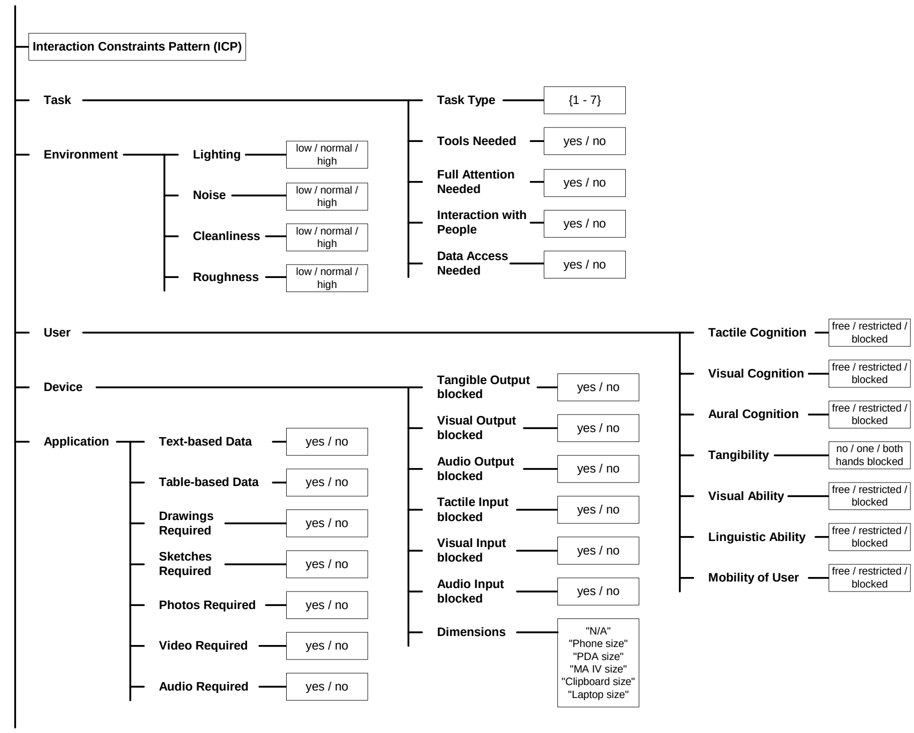

Fig. 2. Possible attribute values for ICPs

Proof-of-concept

A proof-of-concept implementation, the *Interaction Constraints Evaluation Tool (ICE-Tool)*, based on the *Interaction Constraints Model*, illustrates with real-world examples that matching work situations based on the constraints that impact these situations is a valid and workable approach. *ICE-Tool* demonstrates the concept and illustrates the necessary steps to identify work situations with similar work situations.

The following are two examples where *ICE-Tool* could map situations that the designer would probably not have identified due to the difference in their domains.

2.4 Examples

Example 1:

The first example shows a pair of work situations, which a designer might not have thought of or derived from the available literature. It illustrates that the same set of constraints can be found in different applications from different domains. Although the tasks of “comparing a product catalog with the shelf content” (CISS-VR) and “retrieving tourist information in a bar or restaurant” (Deep Map) [11] seem to differ greatly, the interfaces to deliver the multimedia data and the constraints resulting from the work location match fairly closely.

In this example, the match of the environmental constraints and the transfer from one domain to the other domain made it unlikely to imagine the match without *ICE-Tool*. The conditions in an industrial supply room are surely not the same as in a bar or restaurant, but they impact the design of a wearable computer system with the same set of constraints, i.e. the same ICP. Basically, in both work situations we find a low lighting level, a rather noisy environment, and find search and guidance applications being used. Furthermore, in neither situation are the user’s hands blocked during the completion of the task.

Another interesting finding about the Deep Map project is that using the system in a museum restricts the “Linguistic Ability” of the user and the “Audio Input” of the device. However, these restrictions do not result from the high ambient noise, which does not occur at a museum, but in the silence that is expected from museum visitors, which does not allow for using speech input by the user.

Example 2:

In the second example, the system mapped the ICP of a task from a landfill monitoring system (MobileDCT by Meissner [3]) to the constraints of an application in which an audio-only wearable computer (a pick-by-voice system) supports a worker in a distribution center [12]. The landfill inspector uses a GPS system to get guidance to the next measurement point. The worker in the distribution center is fulfilling customer orders from shelves in different aisles. This example illustrates again that the constraints in the different categories can result from different *influences*. In the case of the landfill monitoring system, the visual output is blocked, because the inspector has to walk about an uncovered, cluttered landfill, which does not allow for checking any kind of display – without having to interrupt the primary task. During the order picking, the worker used an audio-only device, which implies a blocked visual output for the device. In both cases, the workers were not able to use their hands for the primary task (taking a measurement and handling groceries, respectively).

The actual benefits in using *ICE-Tool* for the design process result from the possibility to match an identified ICP to a set of constraints that occurred in a work situation of a previously conducted project, and thus to retrieve usability information for the different user interface components in that application. In this way, it is possible to retrieve information about previous projects that did not appear similar to the current project, and were thus not considered as examples for the current design without using *ICE-Tool*.

3 Industry: Speech Technologies in Recent Years

Commercially available speech technology has improved significantly in recent years and has made its way into industrial applications as well as into information systems and consumer products. We find speech recognition and synthesis in car navigation systems, service telephone hotlines, mobile phones, and even in the latest Sony PlayStation® Games [13]. All the above examples are using the "Command & Control" mode of speech recognition, which means that the systems recognize only single words or word sequences from a predefined context.

There are examples of "Dictation Mode" speech recognition as well, but those tend to be applications from certain domains with a highly specialized vocabulary, such as legal or medical applications. We can identify two main reasons for the success of dictation applications in these fields: first, there is a huge number of users sharing the same specialized vocabulary, and second, those users are familiar with (and depending on) dictating – either on tape or directly to an assistant, and thus they are more patient with this technology.

In general though, "Dictation Mode" is not yet to the point where it can be used reliably. It is still not possible to enter free form information about a bridge inspection or evaluate the state of machinery. However, if the application allows the designer to narrow down the possible user input in a way that there is a small set of recognizable items, the use of "Command & Control" gives enough design space to realize a speech-controlled system with satisfactory performance.

In our experience, a "Command & Control" approach with an intelligent management of the active vocabulary is the best solution available today. The goal must be to analyze the tasks to be supported in detail, to only offer the information that the user needs and to only allow the input that the system requires at one point. There are workarounds for free-form textual input, such as providing a selection list of predefined text blocks or even with recording voice utterances. And for most of the applications into which industry is looking at the moment, this procedure is quite acceptable.

The feedback we get from industry when we introduce wearable computers to their application area is quite positive and companies are willing to field-test speech-controlled wearable computers for their tasks. Their motivation at this point is two-fold: first, it is the classic hands-free interaction with the computer while performing other tasks, and second, some of their personnel either oppose textual input, or the companies hope to avoid spelling errors and thus improve automated post-processing of the collected data. From another perspective, there are two user groups to support: the highly trained specialized personnel that need to be time-efficient and thus need to be freed from time-consuming data entry tasks, and the untrained workforce that need IT support to avoid lengthy and costly training sessions.

Focussing on the device, we also see a tendency in the acceptance of speech-controlled wearable computers for industrial applications. Speech-control is in most cases only desired if a head-mounted display, or no display at all, is used. Wearable computers with handheld flat-panel displays are now being replaced by Tablet PCs. Exceptions to this trend are systems with advanced displays that are indeed all-light readable or thin client web pads that are used as wireless display options for wearable computers.

4 Summary & Outlook

Speech-controlled wearable computers are still emerging tools for tasks that require hands-free IT support. However, unless speech technology becomes more reliable for a broader range of recognition modes (i.e., beyond Command & Control), they can only be task-specific tools that are optimized for use in certain applications. However, this is the way that many IT systems – hardware and software – have grown, from task-specific niche products to general tools. What is needed for this transition is more applied research and the close cooperation between universities and industry. The research conducted so far and the products that are commercially available enable us to build acceptable proof-of-concept systems for industrial use. Based on these first steps, we can then focus the research and development on improving and optimizing these systems.

References:

1 www.ce.cmu.edu/~wearables retrieved 04 Nov. 2003

2 Bürgy, Christian; *An Interaction Constraints Model for Mobile and Wearable Computer-Aided Engineering Systems in Industrial Applications*. Thesis, Dept. of Civil and Environmental Engineering, Carnegie Mellon University, Pittsburgh, PA, USA, 2002.

3 Reinhardt, Jan; Garrett, James H. Jr; Scherer, Raimar J.; *The preliminary design of a wearable computer for supporting Construction Progress Monitoring*; International Conference on the Application of Computer Science and Mathematics in Architecture and Civil Engineering (IKM 2000); Weimar, Germany, 2000.

4 Meissner, Iris; *Development of a Concept for the Use of Mobile IT Devices for On-Site Data Collection Concerning Landfill Monitoring and the Prognosis of Landfill Reactions*; Thesis, Technical University, Darmstadt, Germany, 2001.

5 Sunkpho, Jirapon; Garrett, James H. Jr., Smailagic, Asim; Siewiorek, Dan; *MIA: A Wearable Computer for Bridge Inspectors*; Proceedings, IEEE, The Second International Symposium on Wearable Computers, Pittsburgh, PA; 1998.

6 Bürgy, Christian; Garrett, James H., Jr; Klausner, Markus; Anlauf, Jürgen; Nobis, Günter; *Speech-controlled Wearable Computers for Automotive Shop Workers*; Proceedings to SAE World Congress, Detroit, MI, USA 2001.

7 Markowitz, Judith; *Using Speech Recognition*; Prentice Hall; Upper Saddle River, NJ, 1996.

8 Bernsen, Niels Ole; Dybbkjær, Hans; Dybbkjaer, Laila; *What should your computer say?*; IEEE Computer, Vol. 10, No. 12, December 1997.

9 Bredenfeld, Ansgar; Ihler, Edmund, Vogel, Oliver; *GENVIS - Model-Based Generation of Data Visualizers*; Technology of Object-Oriented Languages and Systems (TOOLS 33), St. Malo, France, 2000.

10 Shirogane, Junko; Fukazawa, Yoshiaki; *Method of User-Customizable GUI Generation and Its Evaluation*; 5th Asian Pacific Software Engineering Conference, Taipei, Taiwan, 1998.

11 Malaka, Rainer; *Deep Map: The Multilingual Tourist Guide.* Future Services For Networked Devices (FuSeNetD) – Workshop, Heidelberg, Germany, November 8-9, 1999.

12 http://www.vocollect.com retrieved 04 Nov. 2003

13 http://www.scansoft.com/games/ps2/ retrieved 04 Nov. 2003