Supporting Domain Experts in Determining Viable User Interface Designs for Wearable Computers Used in AEC Work Situations

Christian Bürgy, Wearable Consult, Unterer Burggarten 17, 69221 Dossenheim, Germany
(christian.buergy@wearable-consult.com)

James H. Garrett, Jr, Carnegie Mellon University, Department of Civil and Environmental Engineering
5000 Forbes Avenue, Pittsburgh, PA, 15213-3890, USA
(garrett@cmu.edu)

Summary

The design of mobile IT systems, especially the design of wearable computer systems, is a complex task that requires computer science knowledge, such as that related to hardware configuration and software development, in addition to knowledge of the domain in which the system is intended to be used. Particularly in the AEC sector, it is necessary that the support from mobile information technology fit the work situation at hand. Ideally, the domain expert alone can adjust the wearable computer system to achieve this fit without having to consult IT experts. In this paper, we describe a model that helps in transferring existing design knowledge from non-AEC domains to new projects in the construction area. The base for this is a model and a methodology that describes the usage scenarios of said computer systems in an application-neutral and domain-independent way. Thus, the actual design information and experience will be transferable between different applications and domains.

1 Introduction

In recent years, computers have become smaller - from mainframes to personal computers - and eventually portable, such as laptops. Some of these portable computers are even mobile in that they can be used during transport. The first laptops had to be placed on a solid surface and not moved during usage, whereas today's machines can be carried and simultaneously operated. Since the operation of a standard laptop while walking or working is quite impractical and somewhat dangerous, the wearable computer was invented (Thorp 1998). Wearable computers are worn on a belt or carried in a pocket and have computing power that is equivalent to that of standard laptops. The important differences are the user interfaces that enable usage "on the move," such as head- and body-mounted displays, keyboards, mouse devices or speech recognition and speech synthesis technologies. Figure 1 shows a few examples.

Figure 1: Derivatives of traditional user interfaces. From left to right: L3 Systems WristPC attachable keyboard, Handykey Twiddler, Finger Trackball

There are two usage patterns for wearable computers. One is where the computer is used as an extension of the human. The device is always ready to respond to information retrieval and storage requests, and places address books, calendars, and other personal information within the user's view (Mann 1999). The second usage pattern is that where the computer is used as a task-specific tool that helps workers to perform certain tasks of their job and is not always on but invoked when needed (Billinghurst 1999).

In contrast to wearable computers that evolved from powerful, bigger devices, another group of mobile devices emerged from less powerful, smaller devices. Starting as organizers that could manage a handful of addresses and appointments, those handheld IT devices, now called personal digital assistants (PDA) and Pocket PCs, became more powerful and are currently capable of running similar, or even the same, operating systems as on standard computers. Thus, these devices are about to dominate the field of the mobile computers with the advantage of being developed solely for mobile use. However, due to current limitations in processing power, we cannot implement the same user interfaces as for wearable computers. Eventually, these devices might all merge together through a process referred to as "convergence". In this convergence process, laptops, PDAs, pagers, and mobile phones merge into one mobile or wearable device. Thus, wearable computers should not only be seen as replacements for the laptop, but also as a means for communicating with colleagues, help desks, customers, and keeping track of project management data, such as scheduling, address books, and knowledge bases.

The great challenge for mobile IT devices is to design them for mobile use, i.e. for using them while walking or working. Many mobile input devices have been developed for use "on the move", such as mobile, body-worn pointing devices or keyboards, scanners, or data gloves. But these interactions still involve using at least one hand. Some tasks, however, have to be performed using both hands, which makes the manual operation of an IT device distracting or even impossible. This is especially true for industrial applications, where the targeted users of mobile and wearable computers are workers conducting inspections, doing maintenance, or performing repair jobs (Najjar 1997), (Siegel 1997), (Ockerman 1998). Most of the time, these people have to use both hands for their primary task, which is their actual job, and thus cannot use their hands to operate the device. Thus, hands-free input devices would be ideal for this human-computer interaction (Espisito 1997), (Van Dam 1997), (Billinghurst 1998), (Bass 2000).

2 Interaction Constraints Model – Preliminary Assumptions

To better understand the problems involved in designing wearable computer systems, we 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 2). 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 focused on building a model of all these tasks to be applied during the design process of wearable computers.

Figure 2: 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 wearable computers 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 previous 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*.

3 Interaction Constraints Model

The developed *Interaction Constraints Model* (Buergy 2002) 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 wearable computer 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.

3.1 Interaction Constraints Patterns

Constraints patterns that influence the interaction between the user and the wearable computer system, which we call *Interaction Constraints 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 3.

3.2 Constraint Categories

"Constraints" are the essential component of the model. It contains the information about what the nature of the constraints is, such as a not readable display, and the influence from a specific work situation and a specific work activity, which describe the actual cause of the constraint. A constraint can be defined as "a restriction on the degree of freedom we have in providing a solution" (Leffingwell 2000). This "solution", in the case of the *Interaction Constraints Model*, represents the applicability of certain user interaction means. We categorize constraints in five constraint categories: *User*, *Environment*, *Task*, *Application*, and *Device*. In mapping the constraints to these categories, the resulting restrictions on the user interface design become more obvious and reproducible. This mapping allows for example to map the influence of "user shall wear protective gloves" to a constraint of "user's sense of touch is restricted".

3.2.1 User

User constraints for the interaction with a device are typically input or output constraints that do not allow or at least restrict interaction through specific information channels. Information channels are in this case the human senses. Thus, we can identify the attributes that describe constraints on a user's information channels or human senses in a specific work situation in respect to the user's visual, aural and tactile input and output abilities.

Example: High ambient noise can restrict or block the user's aural cognition and linguistic ability; carrying a flashlight blocks one of the user's hands; and a huge, heavy device restricts the user's mobility.

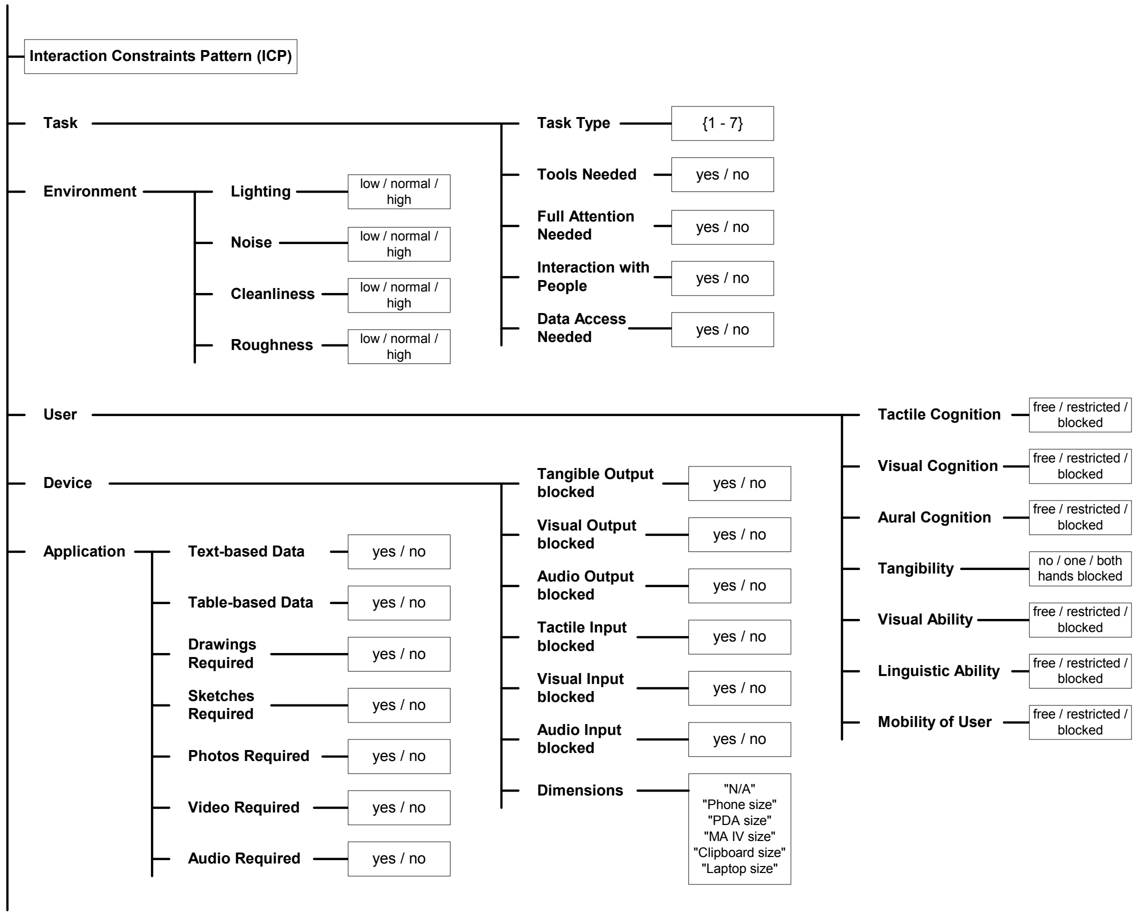

Figure 3: Possible attribute values for *ICPs*

3.2.2 Application

Application constraints are constraints that are mainly based on the kind of data (representation) that has to be entered, accessed, or managed. The collection of these kinds of data will take place in observations of the user's actions in the task analysis, or during the analysis of existing software applications and legacy software.

Example: If an application needs the sketch-based functionality, such as to point out the location of a crack in a bridge component, the sketch-based data attributes is set to "yes"; similarly, the audio-based data attribute is set to "yes" if short text messages from an inspector should be captured as audio files (see Figure 3).

3.2.3 Device

Device constraints are similar to the constraints that affect the user, just at the other end of the information chain. Similar to the user side, we can define visual, aural and tactile input and output channels. The interaction capabilities of a device are more error-prone than the capabilities of the user. This is based on the lack of adaptivity and anticipation the device can offer. A user can more easily compensate a lack of a piece of information.

Example: If the device that is intended for use in a specific application does not offer any sound output, the acoustic output attribute is set to "no"; if the considered device is the size of a PDA, the physical dimensions attribute is set to this value.

3.2.4 Environment

Environment constraints originate mostly from the fact that the mobile or wearable device is used under non-optimal conditions. This means that, for example, the lighting on a bridge might be too bright, caused by sunlight, and in a tunnel too dark, caused by the nature of a tunnel. Note that the sunlight itself is not the constraint, but the resulting bright light in the environment.

Example: Running engines on a construction site can cause high-pitched noise and thus set the ambient noise to "high"; while pouring concrete, the cleanliness level of the construction site can be described as "low".

3.2.5 Task

Task constraints depend mainly on the type of the task. Other task constraints originate from the task's nature and the way it occupies the user or the device.

Example: During a surveying job the surveyor has to carry a rod and chain and other instruments, which would assign the additional tools attribute to "yes"; if a task is difficult and does not allow the user to be distracted, the task needs the user's full attention.

4 Interaction Constraints Evaluation Tool (ICE-Tool)

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. We built *ICE-Tool* as an MS Access Database that contains information about 15 projects and 300 work situations. *ICE-Tool* allows for entering work situations, defined by a work activity and a work location. Furthermore, one can enter the given constraints for that work situation and the related information about the user interfaces that were used in that specific work situation. Thus, the user can retrieve work situations from this case base that serve as design examples from other projects and are not necessarily from the same domain. The following sections describe how *ICE-Tool* is used and which results it gives back to the user.

4.1 Usage Scenario

To demonstrate the concept behind *ICE-Tool*, we present a brief example of how the interaction design of a new wearable computer system can be supported by using the model: first, the system designer performs a task analysis and identifies the work locations and the work activities that occur for the envisioned application. For each relevant combination of work location and work activity, the designer defines a work situation and enters estimated or measured constraints for each work situation into *ICE-Tool's* database. Depending on the number of cases entered in the case-base and the query capabilities of the implementation of the model, the designer gets a set of similar work situations that occurred in previous projects. Now the designer can retrieve information about the user interfaces used in these work situations and evaluate the performance of these user interfaces. Based on that information, the designer can decide which user interface to include in the new system design and which interfaces would not perform well. After collecting user feedback on the design, the designer enters information about the new design and thus adds information to the case-base.

4.2 Usage Example

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.

4.2.1 Example 1

The first example compared a progress-monitoring task on a construction site (Reinhardt, et al. 2000) with a vehicle inspection performed in the field - VuMan Amphibious Vehicle Inspection System (Smailagic, et al. 1998). The similarity of the *ICP* for the two work situations results from the fact that both locations are outside in sunlight, with noisy machinery close by, low cleanliness due to the construction site or vehicle oil, respectively, and rough conditions under which the devices are used for the inspection.

4.2.2 Example 2

In the second example, the system mapped the *ICP* of a task from a landfill monitoring system - MobileDCT (Meissner 2001), 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 (Vocollect 2004). 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.

4.2.3 Benefits

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*. This is especially important in the AEC domain since new technologies adapt slower to the rough conditions present at construction sites and the different tasks that have to be performed during construction processes. The fact the each building is unique – in contrast to mass manufacturing, such as the production of automobiles – makes it even more desirable to re-use existing design knowledge from other domains. The *Interaction Constraints Model*, and *ICE-Tool* based on it, show that we can indeed re-use this knowledge and thus facilitate and speed up the interaction design for wearable computers.

5 Outlook

To use the *Interaction Constraints Model* in future decision support systems, some issues that occurred during the evaluation of the model have to be solved. Mainly, there are three aspects that should be addressed: 1) a refinement of the data model so as to support future use of data analysis methods; 2) the collection of more real-world data; and 3) an advanced version of the

ICE-Tool implementation. The first two issues deal with the problem of retrieving and storing data of previously conducted projects in a right level of detail. The case-base of the system needs to have enough detail to distinguish the work situations in a way that they can be described well. On the other hand, it should not have as much detail to make possible matches between work situations unlikely. Thus, with more projects available in the future, we can run more queries and test and evaluate the system even more to decide which level of detail will be appropriate. The project descriptions given in literature were not detailed enough to have such a broad level of evaluation to date. *ICE-Tool* itself could be improved by more advanced queries that adapt to the refined data structure. We could also think of *ICE-Tool* as being a central data source that can be accessed by the community and serve as a case base where developers from different domains enter their project information and thus share a growing number of work situations. Therefore, the system should be platform-independent, such as a web-based application on a central server. Currently, we are seeking follow-up funding and opportunities to further advance this research.

6 References

Bass, Len (2000); Interaction Technologies: Beyond the Desktop; Chapter in User Interfaces for All. Lawrence Erlbaum Associaties, pp. 81-95.

Billinghurst, M., J. Bowskill; Nick Dyer; Jason Morphett (1998); An Evaluation of Wearable Information Spaces. Virtual Reality Annual International Symposium, Atlanta, GA, USA.

Billinghurst, Mark; Thad Starner (1999); Wearable Devices, New Ways to Manage Information. IEEE, Computer, Vol. 32, No. 1, January 1999. pp. 57-64.

Bürgy, Christian (2002); 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.

Espisito, C. (1997); Wearable Computers: Field-Test Results and System Design Guidelines. Proceedings Interact '97, Sydney, Australia.

Leffingwell, Dean and Don Widrig (2000); Managing Software Requirements – A Unified Approach. Addison-Wesley. p. 44.

Mann, Steve (1999); Humanistic Intelligence: WearComp as a new framework for Intelligent Signal Processing. Proceedings of the IEEE, Vol. 86, No. 11, November 1998, Pages 2123-2151.

Meissner, Iris (2001); 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.

Najjar, L. J., J. C. Thompson, and J. J. Ockerman (1997); A wearable computer for quality assurance inspectors in a food processing plant. In Digest of papers. First International Symposium on Wearable Computers pp. 163-164. Los Alamitos, CA, USA: IEEE Computer Society.

Ockerman, J., J., Thompson, J. C., and Najjar, L. J. (1998); Wearable computers for manufacturing applications. In Proceedings of Flexible Automation and Intelligent Manufacturing.

Reinhardt, Jan, James H. Garrett Jr., Raimar J. Scherer (2000); 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.

Siegel, J. and M. Bauer (1997); A field usability evaluation of a wearable system. In Proceedings of The First International Symposium on Wearable Computers, Cambridge, MA, USA, October, 1997, pp. 18-22.

Smailagic, Asim (1998); An Evaluation of Audio-Centric CMU Wearable Computers. ACM Journal on Special Topics in Mobile Networking and Applications, Vol. 6, No. 4, 1998, pp. 65-76.

Thorp, Edward O. (1998); The Invention of the First Wearable Computer. Proceedings of the Second International Symposium on Wearable Computers (ISWC), Pittsburgh, PA, USA, October, 1998. pp. 4-8.

Van Dam, Andries (1997); Post-WIMP User Interfaces. Communications of the ACM, February 1997, Vol. 40, No. 2, pp.63-67.

Vocollect (2004); http://www.vocollect.com (04 April 2003)