Wearable Computers: An Interface between Humans and Smart Infrastructure Systems

Dipl.-Ing. Christian Bürgy

Prof. James H. Garrett, Jr.

Carnegie Mellon University, Pittsburgh, PA, USA

1. Introduction

'Smart Infrastructure Systems' are bridges, roads, dams, etc. that are capable of monitoring themselves and providing information about their internal structural and material health. Thus, these infrastructure systems will offer valuable data that inspectors will be able to access and use for evaluating their status.

Monitoring and inspecting infrastructure systems are tasks as old as construction itself. Depending on the phase in their life cycle, these infrastructure systems have to be maintained, repaired, refurbished, or reconstructed. To ensure current and up-to-date information about each piece of infrastructure, inspectors have to go and visit each part of the system to record the condition of each component. Modern sensing technologies allow for constant monitoring and aid inspectors in providing histories of recorded data. This helps a) in receiving alerts directly from sensors at the infrastructure itself that measure unusual behavior, and b) in being able to draw additional data from construction elements as needed during the inspection process. Thus, inspection processes and the frequency of inspections can be adapted to fulfill higher information needs, especially for fracture-critical structures, where sudden failures are a big threat.

To ensure the data flow between inspectors and *Smart Infrastructure Systems*, technicians will need an interface to communicate with the different artificial information sources: e.g. wirelessly connected sensors, weather stations, or traffic counters. Mobile computers, such as laptop computers and hand-held Personal

Digital Assistants (PDAs) are possible interfaces, but they do not allow for hands-free operation (see Figure 1). Speech-controlled wearable computers on the other hand, are worn on a belt or stored in the user's pocket and enable hands-free operation. Having both hands free will allow inspectors to ensure two things: especially in dangerous situations, such as climbing a bridge or moving on scaffoldings, they can ensure their own safety by having a safe grip and being able to give their attention to the (dangerous) environment; and the fact that the device is attached to the user's body prevents the possibility that it might get dropped on the floor.

Figure 1: Clockwise from top right: Acer laptop; touchscreen, CPU unit, and battery of Xybernaut MA IV wearable computer; Sharp handheld; Compaq iPaq and Casio Cassiopeia PocketPCs; and Siemens Mobic Pentablet PC.

In this paper, we describe how wearable computers can be used for data collection processes within *Smart Infrastructure Systems*; we focus on the interaction between the inspection personnel, the wearable device, and the embedded sensor devices used in *Smart Infrastructure Systems* (see Figure 2). We can divide this interaction into two segments of interaction: interaction between user and wearable computer and interaction between wearable computer and infrastructure.

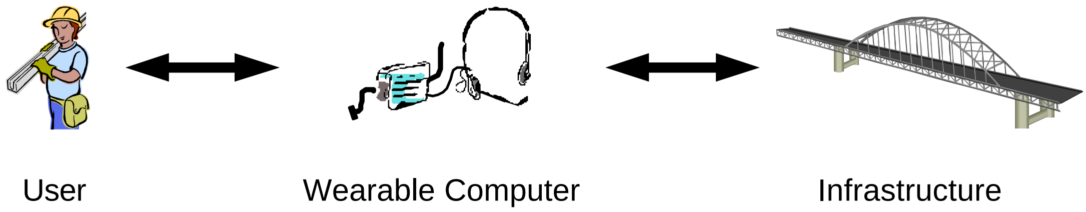

Figure 2: Interaction flow between User, Wearable Computer and Infrastructure

In this paper, wee focus on the first segment, the human (wearable) computer interaction, for which we developed an interaction model (section 2). We also present an outlook for the second segment and the opportunities to access *Smart Infrastructures Systems* by using mobile and wearable computers (section 3).

2. Interaction between User and Wearable Computer

Interaction between two agents usually follows rules and patterns, which is especially true for human-computer interaction (HCI), because the actions of a computer (or a computer interface) are limited. This limitation is caused by the capabilities of the machine itself and the software that is running on the machine. Thus, a computer can only execute actions that are defined and enabled by the software, and the human-computer interaction is limited to these defined actions. The following sections describe an 'Interaction Constraints Model' that we developed to support system designers in their decisions about the "right user interface" for the "right situation".

2.1. Interaction Models

The model within the software that defines and enables actions restricts the interaction and can be referred to as the "interaction model". Beaudouin-Lafon defines interaction models as follows: "An interaction model is a set of principles, rules and properties that guide the design of an interface. It describes how to combine interaction techniques in a meaningful and consistent way and defines the ‘look and feel’ of the interaction from the user's perspective. Properties of the interaction model can be used to evaluate specific interaction designs” [1].

Through mobile and wearable computing, this interaction model has changed compared to the former stationary use of computers, which occurred mainly in office environments. Now that the interaction with a computer was moved away from the desk in an office, we also have to consider other actions that not directly involve interaction with the machine. The primary activity or task of the mobile worker is not to use a computer, but to get the actual job done. The computer can only support this activity. Since dealing with the computer becomes a secondary task, “Direct Manipulation” becomes even more desirable. Users will not accept “the distractions of dealing with tedious interface concepts” [2]. The idea is that not moving a mouse or typing on a keyboard is the actual intension of the user, but to draw a blueprint or write a business letter. Through mobile and wearable computing this idea becomes even more obvious: not carrying around a computer and caring about the ways to interact with it is the intension of the user, but to get support for the actual inspection process or instructions for a complicated installation of a steel structure. Thus, we have to extend the interaction model by activities that are not considered interaction with the machine, but still may influence this interaction. For example, the location and the nature of the activity significantly affect the support that is needed. If this activity occupies the user completely, we might not want to disturb the user with requests from the device; it may be better to postpone the actual interaction with the computer to a later time. Therefore, we should perform extensive task analyses to understand the workflow that we want to support by mobile and wearable computers before we actually design the system.

2.2. Interaction Tasks

The first step to design an IT system that supports workers at their workplace is to look at their actual work and work environment and then find ways to incorporate the IT support into this workflow. The following is our description of different ‘Task’ categories that are defined by their level of interaction between the user and the device. There are three basic categories of ‘Independent Tasks’ that can be identified. Additionally, there are four ‘Composite Tasks’ that are built by combining tasks of the three basic categories. According to Table 1, the *Tasks* are numbered relative to their position in the table:

- ➔ **Primary Task (PT):** No interaction with the device; i.e. no IT support is needed and applied.

- ➔ **Support Task (ST):** Sole interaction with the device and the device supports the user in providing information or accepting input; i.e. ‘productive’ steps are done; e.g. reading a manual, or entering inspection results.

- ➔ **Control Task (CT):** Sole interaction with the device but task only involves navigating the software; i.e. no ‘productive’ steps are done; e.g. scrolling a page or opening a file.

	PT	ST	CT	relevant
1				No
2				No
3				No
4				No
5				Yes
6				Yes
7				Yes

Table 1: *Task* categories in terms of relevance

We only consider *Composite Tasks* No. 5-7 (see Table 1) relevant for our interaction model, since the other categories (*Independent Tasks* and *Composite Task* No. 4) are irrelevant or covered by existing HCI research [3]. This is caused by the non-existence of computer interaction (No. 1), or by the independence from the *Primary Task*, which at least relieves from some of the constraints, although it might not be possible to change the environment completely. In the task analysis, we thus consider tasks, initially classified as No. 2-4 *Tasks* that are constrained by succeeding or preceding *Tasks*, as the corresponding *Task* with PT involvement. For example, if a *Task* only involves a *Control Task* (No. 3), but occurs on scaffolding in high ambient noise caused by a preceding inspection task at a bridge, this *Task* would become a No. 6 *Task* with the constraint imposed by the preceding *Primary Task* (inspection on scaffolding).

2.3. Interaction Constraints

After analyzing the workflow in the task analysis, we have to decide on the "right user interface" for the "right situation". For that, it is important to know whether a specific interaction method or user interface is applicable or whether constraints occur that disallow the use of an interface. In noisy environments, for example, the performance of speech recognition engines is limited, or in other words, the applicability of speech recognition is constrained by the ambient noise.

Leffingwell and Widrig define constraints as "a restriction on the degree of freedom we have in providing a solution" [4]. They mostly describe constraints that are given at the design time, but not those that only occur at the time of the actual use of the software. But the latter are particular interesting for interaction modeling since interaction is highly constrained by the interacting parties and the environment, in which the interaction occurs. An INCOSE (International Council on Systems Engineering) working group goes a bit in that direction in stating that "constraints describe how and where the other requirements apply, or how the work is to be performed" [5]. They proceed with arguing that "most process requirements tend to be constraints" and that some service requirements "may emerge during the project as a result of constraints". Herein, process requirements are defined as "requirements for how the work is to be performed" and service requirements as "requirements for the services that are to be provided". This leads us to the assumption that constraints on the interaction model are requirements at operation time. Examples would be: "climbing a bridge requires both hands"; and "heavy ambient noise disallows use of speech recognition."

Since interaction between users with mobile or wearable computers is highly dependent on such constraints at operation time, our interaction model maps these constraints to the user interfaces. This gives system designers guidance on which user interfaces might be appropriate under the given set of constraints, which is especially difficult with newly introduced user interfaces with which the system designer does not have too much experience.

2.4. User Interfaces

The constraints that occur in each work situation limit the choice of user interfaces that the system designer can apply for that specific situation. Along with the mobility of computers, new human-computer interaction means have been introduced. Most software applications have to date only offered interactions with the computer through traditional interfaces, sometimes called WIMP interfaces, where "WIMP" refers to "Windows, Icons, Mouse, and Point-and-click" [6]. Wearable computing devices and applications often demand a "hands-free" interface, i.e. an interface that the user controls without hands. As indicated before [7], our opinion is that to date, speech technology (speech recognition and speech synthesis) is the most advanced hands-free user interface, and thus the one that most likely will be applicable and acceptable to wearable computers in industrial applications, at least in the foreseeable future.

Most traditional interfaces, such as keyboard and mouse, need a surface on which to place them, and thus are not useful for most mobile computing situations. If mobile and wearable devices offer these interfaces at all, they are mostly derivatives of the office versions, such as the Twiddler, an attachable keyboard, or pointing devices that do not need a flat surface (see Figure 3). Additionally, we see software interfaces as replacements of these hardware components, such as onscreen keyboards.

Other "post-WIMP" interfaces, such as eye-tracking or lip-reading, are developing but are yet to be proven as acceptable in performance (especially on limited hardware resources of mobile and wearable computers). Furthermore, these interfaces have to overcome the same hesitation on the user side that speech recognition had and still has. However, there is a lot of research that will result in new interaction means with wearable computers [8].

Figure 3: Derivatives of traditional user interfaces. From left to right: L3 Systems WristPC attachable keyboard, Handykey Twiddler, Finger Trackball

2.5. Using the Interaction Constraints Model

With the *Interaction Constraints Model*, our approach to facilitating the decision about the right interface at the right situation is to map the possible constraints for different work situation on one side and the pool of user interfaces on the other side. Thus, we map the possible interfaces and their applicability to the constraints, independent from the application or domain in which the constraints occur. Using previous experience with mobile computing projects, we can then seek successful and less successful examples to make more knowledgeable decisions. An example will illustrate this approach:

- A system designer has to design the user interaction, i.e. to choose appropriate user interface components, for supporting a surveying job on a tunnel construction site.
- The *Primary Task* of the surveyor is to measure and document the as-built status of the tunnel to later compare it to the CAD model of the tunnel.
- *Secondary Tasks* and *Control Tasks* are the search and verification of previously documented surveying data that is stored on the IT device.

- The constraints that occur on that site are low ambient (artificial) lighting, limited cleanliness of the equipment, and no wireless connectivity. Constraints from the work activity are that the surveyor needs both hands to handle the equipment and has to watch out while walking about the site.

➔ While entering this set of constraints (which we kept small for the sake of this example), the system that is based on the *Interaction Constraints Model*, suggests that the designer should consider a previous interface that was used in a project in which a bridge inspector who inspected the inside of a bridge abutment could use a handheld display but could also control the computer completely hands-free during the repeated data entry. The *Interaction Constraints Model* hereby represents knowledge about how two interfaces map to given constraints.

This scenario shows how one can use this interaction model, which will become progressively more useful as the data from more projects is entered.

Having a self-contained mobile or wearable computing device mostly helps only to replace the paper-based processes. To actually integrate these devices, in a complete infrastructure monitoring system, we also have to discuss the interaction of the device with other devices that are embedded into the infrastructure itself.

3. Interaction between Wearable Computer and Infrastructure

The second segment of the communication between humans and infrastructure (see Figure 2) is the communication between embedded intelligent sensors and the mobile or wearable computer that the human uses. To replace paper-based processes with mobile and wearable computers is a first step, which led to our approach of first focus on the constraints for the interaction between users and computers. The next step is to design fully integrated infrastructure monitoring systems in which the wearable computer becomes the interface between the human and the actual infrastructure components. Therefore, not only is the interaction between worker and wearable computer important, but so is the interaction between the computer and the sensors within the infrastructure system. However, although this part of the research seems to be only a challenge of integrating different hardware components, it is nevertheless important to be aware of the constraints of this interaction similar to the human-computer interaction. Furthermore, not only does the information exchange between infrastructure monitoring systems determine their usefulness, but so does the information that is provided. The following is an overview of the basic concepts of communication between infrastructure monitoring systems and mobile and wearable computers.

3.1. In-situ data collection

One requirement for *Smart Infrastructure Systems* is that they interact with an inspector or construction personnel by providing meaningful information about their status. Therefore, we have to enable in-situ data collection through embedded or attached sensors and some kind of local data storage that maintains the collected data. This requires sensors that are:

a) capable of collecting accurate and reliable data about specific details of the structure’s status;

b) small enough to not interfere with the structural integrity and the aesthetics of the infrastructure systems; and

c) able to operate throughout the entire observation phase from short-term, such as the curing of concrete, to the entire life-cycle of a building.

The latter requirement imposes some major challenges in the design of these sensors. The main problem, which applies to mobile computing as well, is the power consumption of these systems. We might be able to provide energy for short-term data collection, but if long-term operation is the goal, energy must be either brought to the system, or energy has to be captured at the sensor locations. Systems that are to be embedded into concrete building components should not have wires running into the concrete and thus, the unit has to be completely self-contained. In most cases it will help, if the sensors are able to transmit data wirelessly, either in a short-range communication with another device, or even over a longer-range to central monitoring stations that relay the information to the appropriate personnel.

3.2. Wireless data transmission

Wireless data transmission is most desirable for the transfer of information between mobile and wearable computers and other peripherals or IT devices. In [9], Trupp discusses different wireless communication means for mobile and wearable computers, and states that “wireless connectivity provides an exciting opportunity for the construction industry to extend information management.” Thus, to achieve this extended information management, access to the embedded or attached sensors and other stored information of smart infrastructure systems will most likely be wireless. Two examples of the use of wireless technology for such communication deserve mention: RFID tags and Bluetooth-enabled sensors (Figure 4).

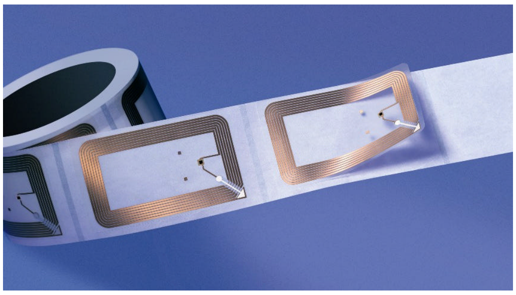

Figure 4a: RFID tags (Rafsec);

Figure 4b: Bluetooth-enabled sensor (Crossbow)

RFID tags (radio frequency identification tags) offer storage capacity and extend the capabilities of bar codes. These small transponders store data and can be activated, accessed, and edited through radio frequency. This allows one to track building components during the construction phase and locate them by using RFID tags that are attached during manufacturing. In that way, the component can track its own history, such as manufacturing date, previously imposed conditions, destination, or earliest loading date. Figure 4a shows an example of self-adhesive RFID tags.

Bluetooth-enabled sensors (Figure 4b) can measure and store status information, such as the stresses or tilt angles of a structure. These sensors store the data they are supposed to capture and – if wired – send the data to the monitoring system, or –

if not wired – store the data and allow monitoring personnel to access this data by means of mobile or wearable computers. Some of these systems even allow for wireless sensor networks that relay the information within the Bluetooth network and are self-contained through enclosed solar power units.

4. Interaction between User and Smart Infrastructure

In the previous sections, we described the two segments of communication between mobile workers using mobile and wearable computers and *Smart Infrastructure Systems* that are capable of monitoring their own status. We showed how to categorize the constraints of the interaction between the user and the mobile computing device, and illustrated the components that are the communication peers of the infrastructure monitoring systems. To build a completely integrated system, we must include *Smart Infrastructure Systems* in the *Interaction Constraints Model* to get the same decision support for the design of the interaction of infrastructure monitoring systems and mobile and wearable computers. Therefore, we will first have to set up some case studies that include emerging wirelessly-connected sensors that demonstrate the capabilities of the sensors and the usefulness of the information these sensors provide.

5. Conclusions

The use of mobile and wearable computers as actual “tools” on the construction site will enable interaction with smart infrastructure systems in the future. But to reach this goal, we face some challenges that have to be solved. The usability and durability of the systems has to be increased, which involves hardware as well as software interfaces. Thus, the design process for the computing systems has to be facilitated to allow domain experts rather than software engineers to design the systems. Based on these new systems, we have to incorporate new work patterns and workflows into the construction process. Furthermore, if more and more immediate information can be collected from the infrastructure, we have to change the way we handle and evaluate this data and the way we use this newly gained information in the design process for the infrastructure itself.

6. References

[1] Beaudouin-Lafon, Michel: Instrumental Interaction: An Interaction Model for Designing Post-WIMP User Interfaces; Proceedings CHI 2000, pp. 446-453.

[2] Shneiderman, Ben: Direct Manipulation for Comprehensible, Predictable and Controllable User Interfaces 1997. p. 3

[3] Kieras, David E. 1997. Task Analysis and the Design of Functionality. The Computer Science and Engineering Handbook. Allen B. Tucker, Jr (editor-in-chief). Boca Raton, Fla, USA. pp. 1401-1423

[4] Leffingwell, Dean and Don Widrig: Managing Software Requirements – A Unified Approach; Addison-Wesley, 2000. p. 44.

[5] George Caple, David Haines, Dave Lamont et. al. Requirements Categorization; Prepared by the Requirements Working Group of the International Council on Systems Engineering. 2001. pp. 4-5

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

[7] Buergy, Christian and James H. Garrett, Jr. "Wearable Computer: Der Entwurf mobiler IT-Systeme zur Unterstuetzung von Arbeitsprozessen im Bauwesen". Fortschritt-Berichte VDI - Junge Wissenschaftler forschen, Reihe 4, Nr. 169, Forum Bauinformatik 2001, Muenchen. VDI Verlag GmbH, Duesseldorf. 2001. pp. 127-134.

[8] Homepage of the International Symposium on Wearable Computers. http://iswc.gatech.edu/ (accessed 20/02/2002).

[9] Trupp, Torsten. *Mobile Computing in Construction Management.* Thesis, Department of Civil Engineering, University of Applied Sciences Giessen-Friedberg, Germany. 2001. p. 7