Situation-aware Interface Design: An Interaction Constraints Model for Finding the Right Interaction for Mobile and Wearable Computer Systems

Christian Bürgy*; James H. Garrett, Jr.*

ABSTRACT

The design of mobile and wearable computing devices involves decisions about how the user interacts with the hardware and software composing the device. Since applications of, and usage environments for, mobile and wearable computers has varied greatly, it has been difficult to build on previously collected design knowledge. The *Interaction Constraints Model* described in this paper offers an application-neutral and domain-independent approach for comparing different applications and usage scenarios. The *Interaction Constraints Model* provides a means to map information about user interface implementations to specific work situations. In this way, a system designer can use a set of generic interaction constraints to identify and retrieve information about user interface components from previous projects. In a proof-of-concept implementation of the *Interaction Constraints Model*, we were able to validate the approach of the model and we demonstrate the usefulness for the design of wearable computer user interactions.

KEYWORDS

Design Constraints; Human-Computer Interaction; Industrial Applications; Mobile Computers; Wearable Computing

1. INTRODUCTION

On construction sites, we see ever changing 'work situations' that differ in their 'work locations' and 'work activities'. Mobile and wearable computer systems can support workers in these *work situations*. But to be useful tools, these systems need to offer specific user interfaces that are appropriate for the *work location* and the *work activity* at hand. The 'Interaction Constraints Model' aids system designers in choosing the right interaction means for specific tasks with respect to the environment in which the task is performed and the kind of mobile or wearable computing system that supports this task.

In this paper, we describe the underlying concept of the *Interaction Constraints Model* and the benefits of using it for interaction design and conclude with showing the result of our proof-of-concept implementation.

2. BACKGROUND

For computer-aided engineering applications, mobile IT support helps to improve construction processes [1] and enables mobile workers to perform their tasks better, faster, and with higher quality, i.e., with higher data consistency (less manual data entry and reentry), shorter data access times (connection to the company's intranet and to online manuals), and better communication means (Internet telephony, short messages, expert forums). However, mobile workers usually perform several different tasks in ever changing environments. This generates different 'constraints' on the system design of the mobile IT support with respect to: the kind of the ***task*** to be performed; the ***application***, for which the task is performed; the influences caused by the ***environment***; the ***device*** chosen as the supporting hardware platform; and the abilities and work patterns of the ***user***.

* Department of Civil and Environmental Engineering, Carnegie Mellon University; Porter Hall 119, 5000 Forbes Ave., Pittsburgh, PA, 15213, USA, Tel.: ++1 (412) 268-5674; e-mail: buergy@cmu.edu; e-mail: garrett@cmu.edu.

Each situation demands that the *user* and the mobile or wearable computer *device* adapt their interaction with respect to the *task*, the *application*, and the *environment*. Many mobile input devices have been developed for use "on the move", such as mobile, body-worn pointing devices and keyboards, scanners, or data gloves. However, these interactions still involve using at least one hand. Some tasks, however, have to be performed using both hands. Thus, it would be helpful to get support in making the decision about which user interface to use for which situation.

3. INTERACTION CONSTRAINTS MODEL

The concept of the *Interaction Constraints Model* is a generic description of different *work situations* based on the *constraints* that impact interaction with mobile and wearable computers in industrial applications. The idea is to compare different *work situations* between different applications of the same or different domains. Thus, we can compare the system design of a mobile IT device and re-use the design decisions. This concept helps to identify similar situations and evaluate how well a specific means of interaction performed in previous applications.

The *Interaction Constraints Model* builds on two definitions: the *constraints* themselves (section 3.1) and the *work situation* that is described by a specific set of *constraints* (section 3.2). A description of the implementation of the model (section 3.3) and a usage example (section 3.4) demonstrate the concept of describing a *work situation* with a set of *constraints*. Section 4 illustrates the contributions and results of the *Interaction Constraints Model*.

1.1 Constraints

Leffingwell and Widrig define constraints as "a restriction on the degree of freedom we have in providing a solution" [2]. Constraints in the *Interaction Constraints Model* restrict the use of a specific interaction modality for a system design for a specific usage situation. Before the introduction of mobile computing, the *constraints* that implied the design of IT systems could be associated with three *constraint categories*: namely "user," "device," and "application." Now there are five categories, since mobile and wearable computers imply changing "tasks" in changing "environments." These *constraint categories* contain *constraints* that influence *constraints* of their own category as well as *constraints* of other categories during operation. For example, a *device constraint*, such as the absence of a display, influences other *device constraints*, such as the need to provide alternative output means; it also influences the *application constraints*, such as no GUI interface being possible. Sections 3.1.1-3.1.5 describe these five *constraints categories* [3].

3.1.1. Task

Tasks are considered to be "states in the working process" as a part of the workflow. *Task constraints* are all those *constraints* that restrict the interaction between the *user* and the *device*, such as a *task* that requires both hands of the *user*.

3.1.2. Environment

Environment constraints are defined as *constraints* of the working / usage *environment* of the *device*, composed of such influences as ambient noise level, lighting, potential hazards (need for gloves, masks, etc.). However, properties of the IT infrastructure are covered by the *device* description.

The considered *environments* are mainly those in which multiple (non-traditional) input modalities are applicable and special demands on the *user* are present (i.e. office environments are covered by existing HCI research and thus not the main target of this research).

3.1.3. Application

Constraints of the *application* influence the user interaction by demanding different navigation

/ operation *tasks* of the software, e.g., a CAD application deals with 2D or even 3D drawing navigation, whereas an inspection application deals more with check lists. There are domain-specific applications, such as construction or manufacturing applications and general applications that for example support the "back office" processes. Furthermore, different application structures or software architectures cause different behaviors of the software. Finally, the *application constraint* category holds the actual interface / interaction layer, i.e. the interface to the *user* of the *device*.

3.1.4. User

User constraints result from different cognitive, logical, and physical abilities of *users*, as well as different expertise and experience. *Users* and their capabilities are also constrained by the working *environment*, such as situations that demand special attention or occupy the *user* in some way.

Working habits of *users* should not primarily go into the *constraints* design, since these habits might change completely with the use of the mobile IT support. However, these habits have to be investigated thoroughly to fully understand the *tasks* that have to be supported by the IT device.

3.1.5. Device

The *device constraints* result from the device itself, as well as from other *IT* components connected to the *device.* These *constraints* are for example the presence of different input / output modalities that are more or less appropriate for a given *task*.

1.2 Work Situations

Work situations are uniquely defined by a combination of a *work location*, the place where the *user* of the mobile or wearable *device* performs a job, and the *work activity*, the actual *task* of the *user*. The following are descriptions of the components that enable the comparison of different *work situations* and thus the re-use of design knowledge.

3.1.6. Work Location

Work locations identify the location, and thus the conditions, in which a *work activity* is performed. The reason to have locations as an identifying factor in the model is the fact that interaction with a device is constrained differently at different locations of one project and at locations of other projects.

Example: "Inspecting a bridge structure" and "assembling tubular steel scaffolding," have many conditions in common; e.g., the sunlight, the height of the workplace, safety concerns, etc. Working on a "tunnel construction project" and in a "pit of an automotive workshop," also have similarities: the artificial light (if any) and the dust / oil of the machines or vehicles.

3.1.7. Work Activity

Work activities represent primary *tasks* of the *user*. Primary *tasks* are the tasks that the envisioned mobile or wearable device will finally support. As mentioned above, *work locations* and *work activities* define unique *work situations*. And the motivation for including the *work activity* as an identifier is similar to that for the *work location*. Here, too, the goal is to find patterns of similar *constraints* that result from different activities and to re-use these patterns for design decisions for new *work situations*.

It may seem hard to compare activities from different domains and to find similar patterns amongst them. But the *work activities* themselves will not be compared, but rather the *constraints* and the *constraints'* influences on the user interaction, which originate in these *work activities*, are compared. Thus, we create *constraints* that are not domain-specific and enable a domain-independent model.

Example: It is obvious that some activities, such as "inspecting bridges" and "inspecting vehicles" have similarities, but even the two activities, "determine the inventory of construction material" and "perform quality assurance at a manufacturing facility," can be mapped to a common constraint pattern.

3.1.8. Work Situation

Work locations and *work activities* define unique *work situations*. The link between the location and the activity is the user who literarily brings the *work activity* to the *work location*. This is a new aspect that is caused by having mobile and wearable computers, which enable IT support away from the desktop or kiosk-like terminals. Thus, we have to identify varying sets of conditions or requirements to which the design has to be adapted, and to which future adaptive devices will adapt automatically.

Each *work situation* is unique in a sense that exactly one *work activity* is performed at one *work location*. However, the conditions at different *work locations* and the demands of different *work activities* can have common patterns, and can thus lead to similar *constraints* on the user interaction with a mobile or wearable device.

Example: We can use the two examples above to show the concept of a work situation: "bridge inspections" and "vehicle inspection" differ in their location; so do "assembling steel scaffolding" and "quality assurance" with respect to the activity. However, "inspecting a bridge's interior structure" and "assuring the product quality in a poorly lit manufacturing plant" have similarities in both respects.

3.2.4 User Interface

Finally, the *Interaction Constraints Model* provides information about user interfaces that were implemented and evaluated in previously conducted projects. The system designer can retrieve this information, which is mapped to specific *work situations,* and use it for designing the user interaction for mobile IT devices for a similar *work situation*.

1.3 Implementation

In order to conduct a proof-of-concept of the *Interaction Constraints Model*, we implemented the model as a database that stores all the necessary information about the *constraints* of *work situations* and the user interfaces that were used in about 15 different previous designs of mobile and wearable computer systems. The implementation allows the user to enter the *constraints* of a new *work situation* and to query the case-base. Figure 1 shows the attributes of the different *constraint categories* that describe each *work situation*.

Each of the attributes of the *constraints* can take several values, e.g. "low," "normal" or "high" ambient lighting. We needed such a simple classification, since the documentation of the investigated projects did not provide more detailed data. However, this classification was sufficient for this proof-of-concept.

As a case-base, we collected project information on 15 system designs of our own research group, from other researchers at Carnegie Mellon University, and from the literature [4]. This case-base was diverse enough to illustrate the usefulness of the application and the domain independence of the model, and showed that we can use *constraints* to retrieve similar situations of previous projects.

1.4 Usage Example

To demonstrate the concept behind the *Interaction Constraints Model*, we want to present a brief example on 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* in an implementation of the *Interaction Constraints Model*. Depending on the amount 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. RESULTS

Using the proof-of-concept implementation, we performed several different types of tests. First, we took situations that an experienced designer could map without any help to prove that the system is valid; then we used the system to query the database for *work situations* that could not easily be imagined, but served as good design examples.

One example was to compare a progress-monitoring task on a construction site (Progress Monitor by Reinhardt, et al. [5]) with a vehicle inspection performed in the field (VuMan Amphibious Vehicle Inspection System by Smailagic, et al. [6]). The similarity of the *constraint patterns* 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.

The second example in which the system returned corresponding design examples from a different domain were the transmission of patient data of EMS personnel at a highway accident at night and an inventory maintenance task in a tunnel construction site. The matches derived from the system show that in the different *work situations* of the two applications, the same set of *constraints* restricts the user interaction and thus can be designed in a similar way.

Finally, the system mapped a tourist guide application helping a tourist in a restaurant to find the next attraction in an online multimedia guide and a worker querying a mobile spare part database in a manufacturing application. In this example, the match of the *environment constraints* and the transfer from one domain to the other domain made it unlikely to imagine the match without the help of the *Interaction Constraints Model* implementation. The conditions in an industrial supply room are surely not the same as in a bar or restaurant, but they impact the design of mobile IT system with the same set of *constraints*. Another finding about the tourist guide project is that using the system in a museum restricts the “Linguistic Ability” of the user and the “Audio Input” of the device. Thus, it matches in these categories to many other industrial applications. However, these restrictions do not result from the high ambient noise, which does not occur at a museum, but in the required silence expected from museum visitors, which does not allow for using speech input by the user.

These results showed us that we indeed could compare *work situations* based on the *constraints* that impact the user interaction. However, we have to extend the case-base of mobile and wearable computer design projects to sufficiently support a broader variety of *work situations*.

5. CONCLUSIONS

With the *Interaction Constraints Model*, our approach to determining the best interfaces for a given situation is to map the possible *constraints* for different *work situations* to a set of potential user interfaces. Thus, we map the possible interfaces and their applicability to the *constraints*, independent from the application or domain in which the *constraints* occur. This approach allows a more systematic means by which to search for and apply previous experience gained in mobile computing projects.

6. REFERENCES

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

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

[3] Bürgy, Christian and James H. Garrett, Jr. 2002. *Wearable Computers: An Interface between Humans and Smart Infrastructure Systems.* Bauen mit Computern 2002, Bonn, Germany. VDI Verlag GmbH, Duesseldorf, Germany.

[4] 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. p. 82

[5] 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.

[6] 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.

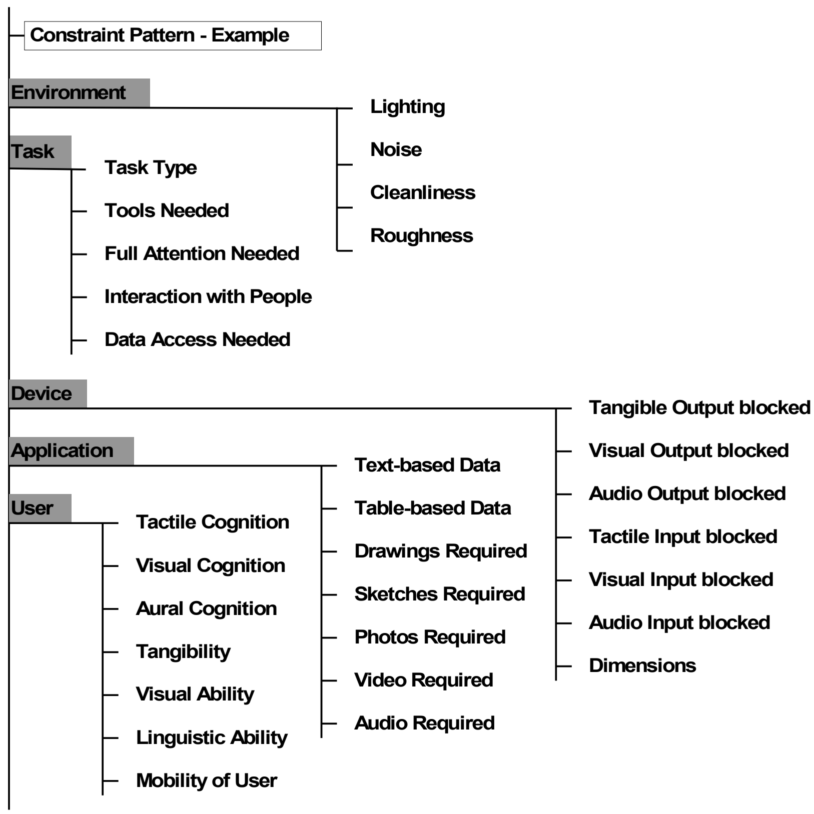

Figure 1: Each *work situation* can be described with a specific set of *constraints* (or *constraint pattern*).