Grouping Mechanisms for Smart Objects Based On Implicit Interaction and Context Proximity

Stavros Antifakos, Bernt Schiele
ETH Zurich, Switzerland

Lars Erik Holmquist
Viktoria Institute, Göteborg, Sweden

ABSTRACT
When everyday objects become equipped with computation and sensors, it will be important to explore interaction techniques that rely on natural actions. We show examples of how non-accidental simultaneous movement of "smart" objects can be exploited as implicit interaction. Applications include implicit access control when opening a door and an automatic packing list creator. This principle of implicit interaction based on non-accidental movement patterns can be extended to other context parameters, forming a context proximity hierarchy.

INTRODUCTION
As defined by Weiser, ubiquitous computing is "invisible, everywhere computing that does not live on a personal device of any sort, but is in the woodwork everywhere." [3] In some ways, this vision could prove to be as much a problem as a solution! When more and more everyday artifacts and environments become augmented with computation and sensing, new problems arise in the design of human-computer interaction, since every object becomes a potential input device. Much like peripheral and ambient information displays have been introduced to lessen the strain of information overload, various ways of background sensing and interaction will need to be developed to avoid potential problems in users' interaction with computer-augmented environments.

One solution to this problem would be to design interfaces based on *implicit human-computer interaction.* This has been defined as "an action, performed by the user that is not primarily aimed to interact with a computerized system but which such a system understands as input." [2] In other words, whereas the user continues to interact with everyday objects as normal, we may use these actions as a sort of "side-effect" to also produce input for a computer system.

We are exploring how we can create implicit interaction with everyday artifacts that are equipped with sensors of various kinds. More specifically we exemplify how *non-accidental movements of objects* can be used to support implicit HCI. By using accelerometers attached to everyday objects, it is possible to detect if two or more objects share the same movement pattern. This information can be used to support everyday tasks, without introducing any additional interaction demands to the user. Other context parameters besides movement can be used in a similar fashion for implicit interaction. We call the resulting principle the *context proximity hierarchy*.

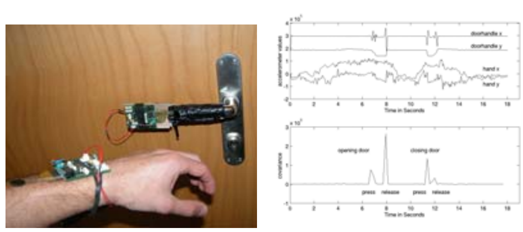

Figure 1: Access Control example, showing the door handle and the person's wrist equipped with accelerometers (left). Acceleration values of the door handle and the persons hand (top right). and the correlation measure use to detect the use of the door handle by a certain person (lower left).

AN EXPLICIT GROUPING MECHANISM: SMART-ITS FRIENDS
Smart-Its Friends [1] is an example of a grouping mechanism based on *explicit* interaction. When a user wants to tell two or more "smart" objects that they belong to the same group, she holds them together and shakes them. Via radio communication, all objects continuously communicate their trajectory, as determined by accelerometers. Since the objects that are shaken together will be the only ones that have the same trajectory, they can use this information to create a grouping.

The underlying principle of Smart-Its Friends uses an explicit gesture – shaking – to group and establish a special relation between objects. This principle has many interesting applications: If you want to be sure that your wrist-watch beeps whenever you leave your cell-phone behind you simply shake them in order to make them "friends". Even though the underlying principle is general and powerful in itself it does require an explicit action from the user. Rather than to rely on explicit interaction this paper explores implicit interaction based on non-accidental movement patterns to establish a special relation between objects.

TWO EXAMPLES OF IMPLICIT INTERACTION BASED ON NON-ACCIDENTAL MOVEMENT

Access Control
Today, many access control systems are installed, so that (restricted) access can be granted to people. Those access control systems usually require an explicit action from the employee such as to swipe an identification badge or to use a specific number key which are prone to be lost or forgotten.

Here, we propose to use the action of pressing the door handle – which is necessary to open the door – to identify the person, and give him the appropriate access. For this we use two accelerometers: one on the door handle and one on the person's wrist (Figure 1). When the person presses the door handle we detect the simultaneous acceleration pattern of the door handle as well as the wrist. By verifying that the owner of the wrist-accelerometer is indeed allowed to access this particular door, the system can grant permission to that person by opening the lock. This is an example of implicit interaction since the only action required from the person is the normal door-opening action namely pressing the door-handle. Figure 1 shows 2D-acceleration data of the door handle and of the person's hand pressing the handle twice. The correlation measure between the signals clearly shows how the pressing of the door handle can be detected.

Automatic Packing List Generation

The task of packing a set of goods into a box and then having to generate a packing list is common in both industry and everyday life. For instance, at a typical Internet retailer, books or other items belonging to an order are packed in a box and an invoice is generated. In other industries mechanical parts, computers, or raw materials are packed and labeled before shipment. Even when moving your household you would be happy to know in which box you packed that fragile set of crystal glasses or some essential piece of clothing.

By attaching accelerometers to the goods, we can record the individual movements of the goods and determine which possess similar movement patterns. The normal action of moving the box around serves as an implicit grouping mechanism of all items in that particular box. The similarity of the movements of those items is again non-accidental since the items packed in the same box will be the only ones that have the same trajectory. Determining the similarity of the movements is therefore sufficient to group those objects, which are packed together. When the objects have been grouped, a packing list of all items can be generated, or other checks on the goods could be performed such as completeness of an order.

Implementation Details

The above demonstrations are based on Smart-Its technology [4]. We used the standard configuration of the Smart-Its sensor board including a 2D-acceleration sensor (ADXL 202). This is combined with a radio frequency communication module, also part of the Smart-Its platform. To decide whether two or more objects are moving together, it is sufficient to calculate the correlation value between the objects acceleration values signals, which gives us a measure of how likely the objects, are to be in the same group. In the demonstrations a Smart-It was attached to each object, transmitting its acceleration values to a central processing unit, which is then responsible of calculating the similarity between the movement trajectories.

CONTEXT PROXIMITY

The detection of non-accidental movements can be viewed as comparing a part of their context. The next step is to compare other contextual information of objects to enable applications where the moving of objects is not realistic or not desired.

Table 1: Context Proximity Hierarchy

Level	Physical characteristics/events
dynamics of the objects	object movement, light changes, …
dynamics of the environment	people moving, light switching on/off, doors banging, …
static state of the environment	"weather", temperature, light level, noise level, …

Table 1 shows a more general approach to classifying different types of physical characteristics for comparing context. We call this approach a "Context Proximity Hierarchy" as the context of two entities can be compared on any of the given levels. In this hierarchy we have classified the movement of the objects in the first level, namely the "dynamics of the objects". All examples presented above draw their context information from this level. When object movement is not available the "dynamics in the environment" can be used to gain knowledge about the situation the objects are in. Here the effects of events such as people moving about in the surroundings, doors being banged, people talking, or lights being switched on and off can be captured by sensors. On the lowest level the comparison of the static characteristics of the environment is modeled. These consist of the physical parameters of the environment such as light, noise level, and temperature, which might be subsumed as "weather" data. They can be used to get a prior about whether the objects might have a similar context or not. This information could for instance be used as a baseline to make the grouping mechanisms more reliable.

CONCLUSION AND FUTURE WORK

We have shown how a basic grouping mechanism can be implemented and used to provide implicit input for everyday tasks. Our current implementation is based on exploiting the non-accidental movements of two or more objects to determine if they are moved together. In the future, as the cost of sensors and communication technology decreases, we will likely see sensors added to a variety of objects and find a multitude of uses for them. In this case implicit interaction techniques such as those presented above might help to decrease the complexity of human-computer interaction in many everyday situations.

ACKNOLEDGEMENTS

The Smart-Its project is funded in part by the Commission of the European Union under contract IST-2000-25428, and by the Swiss Federal Office for Education and Science (BBW 00.0281).

REFERENCES

1. Holmquist, L.E., Mattern, F., Schiele, B., Alahuhta, P., Beigl, M. and Gellersen, H-W. *Smart-Its Friends: A Technique for Users to Easily Establish Connections between Smart Artefacts.* UbiComp 2001, USA.
2. Schmidt, A. *Implicit Human-Computer Interaction through Context*, Personal Technologies 4 (2&3), 2000.
3. Weiser, M. *Ubiquitous Computing* (definition #1). http://www.ubiq.com/hypertext/weiser/UbiHome.html
4. Smart-Its Project: http://www.smart-its.org