Smart-Its Friends: A Technique for Users to Easily Establish Connections between Smart Artefacts

Lars Erik Holmquist[1], Friedemann Mattern[2], Bernt Schiele[3], Petteri Alahuhta[4], Michael Beigl[5] and Hans-W. Gellersen[6]

[1]PLAY Research Studio, Interactive Institute, Gothenburg, Sweden; leh@interactiveinstitute.se
[2]ETH Zurich, Distributed Systems Group, Zurich, Switzerland; mattern@inf.ethz.ch
[3]ETH Zurich, Perceptual Computing and Computer Vision Group, Zurich, Switzerland; schiele@inf.ethz.ch
[4]VTT Electronics, Oulu, Finland; petteri.alahuhta@vtt.fi
[5]TecO, University of Karlsruhe, Germany; michael@teco.edu
[6]Lancaster University, Department of Computing, Lancaster, UK; hwg@comp.lancs.ac.uk

Abstract. Ubiquitous computing is associated with a vision of everything being connected to everything. However, for successful applications to emerge, not the quantity but the quality and usefulness of connections will matter. Our concern is how qualitative relations and more selective connections can be established between smart artefacts, and how users can retain control over artefact interconnection. We propose *context proximity* for selective artefact communication, using the context of artefacts for matchmaking. We further suggest to empower users with simple but effective means to impose the same context on a number of artefacts. To prove our point we have implemented *Smart-Its Friends*, small embedded devices that become connected when a user holds them together and shakes them.

1   Introduction

The drive toward ubiquitous computing gives rise to smart artefacts, which are objects of our everyday lives augmented with information technology. These artefacts will retain their original use and appearance while computing is expected to provide added value in the background. In particular, added value is expected to arise from meaningful interconnection of smart artefacts. Advances in wireless networking will in principle enable interconnection of large numbers of artefacts, but how can more specific relationships be established across such networks? And how can users retain control over information exchange among artefacts? In this technical note we propose to base artefact relationships on the artefacts' context, and describe a very easy-to-use technique for users to explicitly establish artefact connections.

The work we describe was conducted at the early stages of the Smart-Its project on technologies for computer-augmentation of everyday artefacts, inspired by previous work in the Mediacup project [1]. One of the project objectives is to develop a range of small, embedded devices as platforms for augmentation and interconnection of artefacts. These devices, the *Smart-Its*, in general integrate sensing, processing and communication with variations in perceptual and computational capability. Sensors and perception techniques are integrated to facilitate autonomous awareness of an

artefact's context, independent of infrastructure. Wireless communication is added to facilitate the sharing of such context among artefacts. In our earlier work we have explored applications enabled by artefact-based context acquisition and sharing [3]. The research focus in this project is on collective behaviour of Smart-Its-enabled artefacts, such as collective perception of the environment.

One area of investigation is correlation of context across artefacts as foundation for collective functionality. In a sense, such correlation can be used for matchmaking among artefacts, enabling artefacts to discover others with the same or similar context. Hence, the idea is to have connections established based on *context proximity*, which we will discuss further in section 2. As we will point out in that discussion, context may well be a user action, and context proximity may be exploited in explicit interfaces. In section 3 we describe a prototype implementation of such an interface, the *Smart-Its Friends*. With this interface technique, a user can establish a connection between two smart artefacts by simply holding them together and shaking them.

2   Context proximity as Basis for Connection of Artefacts

The traditional approach to connect computational entities over a network is to select them by means of a unique address. In mobile computing environments, new paradigms have emerged to support more spontaneous connection of computing nodes that do not require a priori knowledge of each other. In such a setting, connections are dynamically established based on discovery of all devices within sending range.

Another paradigm that emerges with ubiquitous computing is proximity-based communication, to support connection of temporarily co-located artefacts [6]. The required proximity is often defined by the network technology, for instance infrared, and may be the basis for specific techniques to connect artefacts explicitly, for example *point-and-shoot*. Spatial proximity may also be handled for connections at a higher level of abstraction, for instance for situated communication in which spatial locations are used to relay messages [7]. Spatial proximity can be further generalized to a notion of context proximity. The stick-e-notes system may serve as an example for the use of context proximity, using space and time to establish context-dependent information flow [2].

Context proximity as a paradigm for connecting artefacts is of particular interest in our project, as artefacts enabled with Smart-Its have built-in context awareness. Context in Smart-Its generally refers to information about the system environment obtained through sensors. Context comprises raw sensor data, generic percepts extracted from sensors, and artefact- or application-specific information resulting from further abstraction. Hence, Smart-Its are near to each other in terms of context proximity when they experience similar situations or conditions. We envision two general ways of exploiting such context proximity for connection of artefacts: implicit connection vs. explicit connection.

Implicit connection based on context proximity means that artefacts will be automatically connected if their individual contexts are within certain proximity, depending on a suitable distance metric. For example, artefacts or devices worn by a

person will experience similar context (e.g. same movement patterns) which could be used to establish a private body network that is not easily fooled to provide information access to other devices that may be detected near-by.

Context proximity can be used for explicit user-controlled connection of artefacts, if user actions are employed as context. For example, a user may perform the same gesture on different artefacts, imposing on them the same kind of context. We have explored this interface concept further in a prototype implementation of *Smart-Its Friends.*
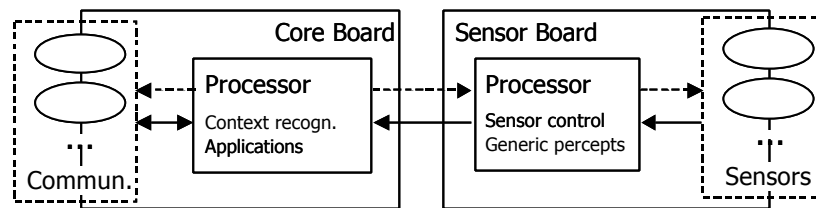

Figure 1. Smart-Its Device Architecture

3   Smart-Its Friends

The Smart-Its devices are based on a modular design with two boards, separating the sensor unit from the core unit. The main components and the data and control flow on the device are illustrated in figure 1. Acquisition of data is allocated on the sensor unit, with a dedicated processor for sensor control and extraction of generic features. Overall device control, application-specific processing, and communication with other Smart-Its are allocated on the core unit. Application-specific processing may for example be computation of artefact-specific context or any other further abstraction from sensor data. The communication interface may support different kinds of network. Generally we assume that all Smart-Its communicate over a shared wireless medium, but some Smart-Its may support additional networks to implement gateways.

The Smart-Its device prototypes that were implemented to explore the Friends concept is shown in figure 2. The two boards have a size of about 4 x 5 cm, and are mounted on top of each other. The device uses PIC micro-controllers on both boards. The sensor unit is further equipped with a two-axis accelerometer. Another sensor, a simple ball switch, is integrated on the core board and directly connected to an interrupt on the core processor. This enables the device to go into an energy preserving mode when no movement occurs, and to wake up instantly on movement. The core board is further equipped with an RFM interface for wireless communication. The communication is based on detection of Smart-Its within sending range. All Smart-Its use RFM as shared broadcast medium, based on a simple carrier sense collision avoidance protocol (CS/CA).
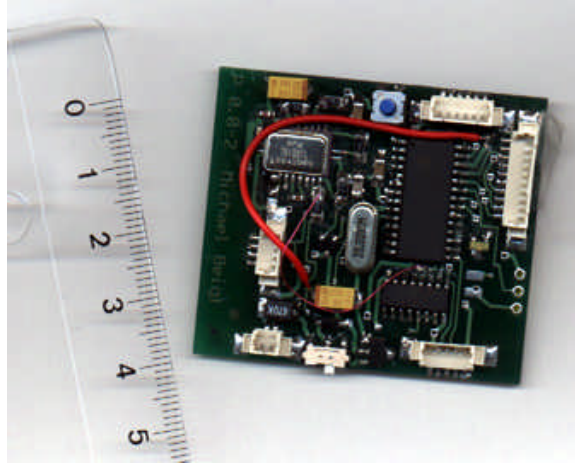
Figure 2. Prototype implementation of the Smart-Its device for validation of the Friends technique for explicit connection of smart artefacts

Connecting Smart-Its Based on Context-Matching

When the Smart-It device is awake, its accelerometer is read with a hard sampling rate of 1 MHz, but only in intervals of short duration at a much lower soft sampling rate to minimize energy consumption. The movement data captured per interval is passed from the sensor unit to the core processor. The data together with the Smart-Its ID is then broadcast over the shared medium to all other Smart-Its in listening range. A Smart-It that receives movement data from another device compares the data to its own most recent movement pattern. If the difference is below a specified threshold then it accepts the other Smart-It as Friend and establishes a dedicated connection. It is up to specific applications, how such a connection would be used.

The connection may physically break when the Smart-Its move out of each others sending range. However, as soon as they are close enough again they would still recognize each other as friend and re-establish the connection. However, depending on the application, it may also make sense that friend's connections are given up over time or when the context changes in certain ways.

Shaking Artefacts to Impose a Connection: an Invisible User Interface

The behaviour built into Smart-Its Friends as described above provides users with a very easy-to-use interface to impose friendship on Smart-Its (or artefacts augmented with Smart-Its). Users can now simply take two devices they wish to connect and move them together, for instance holding them in the hand, briefly waving or shaking. This imposes the same movement pattern on the devices. In all likelihood, this pattern will be different from that measured at the same time in other Smart-Its in the vicinity, ruling out unintended connection. This interface technique is particularly

easy to use, as it does not matter what kind of movement is carried out: it only matters that it is imposed simultaneously on all devices that are to be connected.

It should be noted that in our implementation the orientation of the devices matter. The deliberately simple matching algorithm requires that devices have the same orientation, e.g. held on top of each other or next to each other. If the devices instead are held with opposing orientation, this will be reflected in opposing movement patterns. In our first prototype we have used this for explicit disconnection of devices.

Application of Smart-Its Friends

For our proof-of-concept we have built a simple awareness application for Smart-Its Friends. As soon as a friend becomes connected, the application will notify this to the user with a brief beep. This notification also occurs after a friend has been temporarily out of range and hence disconnected.

This kind of awareness support is similar to that provided by colleague or group awareness devices such as the Hummingbird [4]. In fact, Smart-Its might be considered as add-on units that can be attached to Hummingbird-like devices, augmenting them with the Friends technique. People arriving together at a crowded party and wanting to maintain mutual awareness over the evening would briefly put their augmented awareness devices together and give them a shake. The devices can then notify the users whenever they are close to each other, creating a sense of awareness similar to that of on-line applications such as ICQ.

Another similar application for Smart-Its Friends could be a child monitor: By taking two Smart-Its enhanced objects, e.g. two brooches than can be worn by child and parent, and shaking them together, a connection would be established. Whenever the child strays out of reach, the parent will get a notification.

In a different kind of application domain, one can further imagine to use Smart-Its Friends to connect personal objects. For example, a user may connect their credit card to other personal items such as a car key or pen knife. A smart credit card would then only function if a friend was around, rendering the card useless when it gets lost.

4   Discussion

The Smart-Its Friends technique has obvious application potential for dynamically creating a logical proximity relation and communication channels between artefacts, without having to worry about underlying protocols. In the extension, the method can also be used for other types of end-user programming of smart artefacts. From a user interface perspective the programming and customisation of ubiquitous computing artefacts is often problematic, especially for artefacts where there are no explicit input or output devices. Therefore, it is important to find ways to customise and program artefacts using other natural activities – gestures, sounds, etc. – something that the Smart-Its Friends mechanism achieves.

A natural extension of the concept would be to introduce "modifier objects" that can change the behaviour of other artefacts. By holding a Smart-It augmented object together with a modifier and shaking them, the artefact's behaviour could be changed to that specified by the modifier. One example of such a modifier would be a "magic

stick" with an easy-to-use slider that would allow to parameterise the distance a child wearing a Smart-It is allowed to be away from the parent's Smart-It before an alarm is raised. Another example would be a simple ear-shaped object that might tell another Smart-It to start paying attention to data from its audio sensors – in other words, to start listening – whereas an eye-shaped object might turn on the Smart-It's visual perception. In the child monitor example above, this might mean that the two Smart-Its open an audio channel between each other, so that the parent can hear sounds from the child. This would make it useful for also monitoring the activities of smaller children and babies, where aural information is more likely to be important than movement.

It should be pointed out that "modifier objects" do not really add any functionality that could not be accessed by other means, for instance by adding specialised buttons. Instead they are intended to make interaction simple and more intuitive by acting like physical representations or "tokens" for different functions, much like icons on the computer desktop call up different programs [5]. Making different functions physically manifested in this way might be a way to achieve the invisible or "disappearing" computer interface and give end-users easier access to complicated functionality in ubiquitous computing applications.

Acknowledgements

References

1. Beigl, M., Gellersen, H.-W. and Schmidt, A. MediaCups: Experience with Design and Use of Computer-Augmented Everyday Objects. In *Computer Networks*, Vol. 35, No. 4, Special Issue on Pervasive Computing, Elsevier, March 2001, pp. 401-409.
2. Brown, P.J., The stick-e Document: a Framework for Creating Context-aware Applications. *Electronic Publishing '96*, pp. 259-272, 1996.
3. Gellersen, H.-W., Schmidt, A., Beigl, M. Adding Some Smartness to Devices and Everyday Things. Proceedings of IEEE Workshop on Mobile Computing Systems and Applications 2000 (WMCSA'00), Dec. 2000, Monterey, USA, IEEE Press.
4. Holmquist, L.E., Falk J. and Wigström, J. Supporting Group Collaboration with Inter-Personal Awareness Devices. In *Personal Technologies*, Vol. 3, Nos. 1&2, pp. 13-21, 1999.
5. Holmquist, L.E., Redström J. and Ljungstrand, P. Token-Based Access to Digital Information. Proceedings of First International Symposium on Handheld and Ubiquitous Computing (HUC'99), Karlsruhe, Germany, Springer-Verlag, pp. 234-245, 1999.
6. Hupfeld, F. and Beigl, M. Spatially Aware Local Communication in the RAUM System. Proceedings of Workshop on Interactive Distributed Multimedia Systems (IDMS 2000), Enschede, The Netherlands, October 17-20, 2000, pp. 285-296.
7. Rekimoto, J., Ayatsuka, Y., Hayashi, K. Augmentable Reality: Situated Communication through Digital and Physical Spaces. Proceedings of IEEE 2nd International Symposium on Wearable Computers (ISWC'98), pp. 68-75, 1998.