Requesting Pervasive Services by Touching RFID Tags

Using a mobile phone to touch visual symbols on RFID tags that indicate and trigger intuitive services helps connect the physical and digital environments.

Jukka Riekki and Timo Salminen
University of Oulu

Ismo Alakärppä
University of Lapland

The growth of pervasive computing has introduced an increasing number of services in our everyday environments. We don't just access them at desktop computers but everywhere our activities lead us—using mobile terminals and built-in technology. This transition, while positive, also introduces considerable challenges to discovering and selecting services. The traditional user interface is often too cumbersome, and fully autonomous mechanisms don't give users enough control over a digital environment's behavior.

We propose a general framework for requesting services by touching RFID tags. We focus on using a mobile phone as a mediator between the user and the local environment's services. We provide RFID tags that intuitively relate to the services, creating a natural, calm user experience. The information related to the tag and the mobile phone (for example, location) sets major constraints on the situation. So, we can relax the system requirements for recognizing the user's situation (that is, context) and for locating the proper services. The constraints are especially strong when the user must nearly touch the tag with the phone because we can interpret reading a tag as an intentional action. Users stay in control of the system. However, unlike with completely manual operation, users don't need to know or enter the service parameters after touching a tag (the system can set these automatically).

In this article, we present a general framework for requesting pervasive services by touching tags, representations for the data stored in the tags and the visual symbols shown to the users, a concrete middleware implementation, and usability and user experience studies.

Requesting services by touch

In our framework, RFID tags help bridge the physical and digital worlds, and we use mobile phones as physical objects rather than as traditional I/O devices. Brygg Ullmer and Hiroshi Ishii categorize this as a *relational interface*, in which the system maps logical relationships between physical objects to computational representations.[1] Many researchers have suggested tangible interfaces, including several by Ullmer and Ishii.[1] Others have proposed such interfaces to

- manipulate digital data as physical tiles,[2]
- link physical documents with digital ones,[3]
- identify objects by pointing at them,[4] and
- adjust a mobile terminal's profile by touching RFID tags with the terminal.[5]

Context

If a user can request services from a pervasive system by touching tags in the environment, how does this action activate the right service? The key to a general solution comes from realizing that you can interpret the data read from a tag as contextual information. For example, you can create the context description "User A has requested a calendar view by touching object B," where A and

Published by the IEEE CS and IEEE ComSoc ■ 1536-1268/06/$20.00 © 2006 IEEE

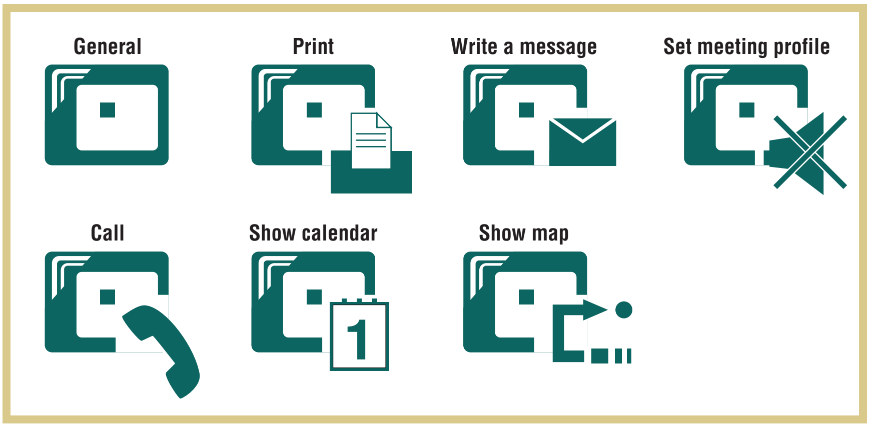


Figure 1. Visual symbols: general and special tags.

B are identifiers when the user touches a calendar symbol attached to an object. So, touch-activated services resemble any other context-aware service.

Obviously, we want the activated service to be the intended one. We can facilitate this matching by using the RFID tag as an index placed in the physical world that points to the right digital service. The visual symbol on the tag communicates to users the service they'll activate. The tag also stores data that triggers the corresponding service when delivered to the system.

The system generates context events from the data produced by the tag reader and other available data. System components implementing the services subscribe to context events that they have an interest in and can process. The mobile terminal and the network contain these components, which can fetch additional information using contextual information as a search key. They can also request services from other components and interact with users through the mobile terminal's and other devices' user interfaces. When alternatives for performing the requested service exist, the system can use additional context information to narrow the choices. For example, with printing, when more than one document is open, the system could always print the one showing on the phone's display. It could also present users a list of alternatives.

Tags

We use two types of tags. A *general* tag identifies the object it's attached to. The visual symbol (General) in the top-left corner in figure 1 indicates general tags in the environment. *Special* tags identify objects, but they also represent additional information related to the object: an action to be performed, a location, a URL, and so on. The rest of the visual symbols in figure 1 are special symbols, with the appearance of the general symbol augmented by an action symbol.

For example, a printer might have three special tags: Print, Contact maintenance, and Info. This would result in a maximum of four tags attached to a printer, including the general tag, which stores only an object identifier identifying the printer. The data stored in a special tag corresponds with its symbol—in this case, each special tag contains data identifying an action. You can also store parameters—for example, the telephone number of maintenance personnel—in the tag or have them fetched from the network using the data read from the tag as a search key. At a minimum, you could store only globally unique tag identifiers in the RFID tags and fetch everything else from the network. However, we prefer locally available information and so use the RFID tags' storage capabilities.

We selected the NFC (Near Field Communication) format for storing data in RFID tags because it supports storing several records on a tag. We aimed to store each data item (object identifier, action, URL, location, and so on) as a separate record. We selected the Electronic Product Code for identifying objects. The EPC contains an organization prefix, an object class, and a serial number. The organization prefix defines the organization responsible for maintaining object classes and serial numbers. The object class determines the object type associated with the tag—for example, *Printer*, *Poster*, *Room*, and *Doorway*. Finally, serial numbers enumerate the different instances of each object class.

So far, we've specified the EPC representation for object identifiers (actions are currently represented as part of the EPC). We can add other data items flexibly as new NFC records.

Middleware

We identified the fundamental requirements for generic-use pervasive middleware by examining existing pervasive systems (for example, Gaia,[6] one.world,[7] and Aura[8]) and requirement specifications. The requirements are

- *interoperability:* the ability of two or more systems or components to exchange information and to use the exchanged information;
- *discoverability*: the ability for system components to discover surrounding entities and, conversely, for other entities in the surroundings to discover the system components;
- *location transparency:* the ability for system component locations to be transparent to other components, the programmer, and the user;
- *adaptability:* the ability of a software entity to adapt to a changing environment; and
- *context-awareness:* the ability to have awareness of the user's context, which enables a pervasive system to provide relevant information and services to users.

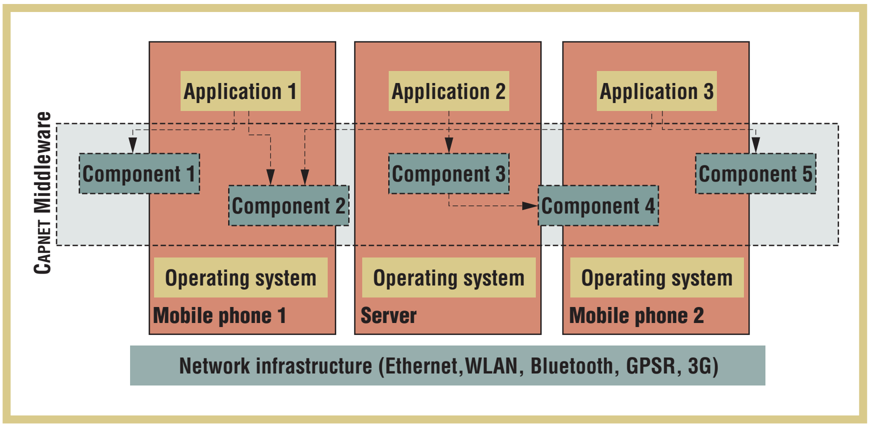


Figure 2. Context-Aware Pervasive Networking architecture overview.

We used these requirements as the basis for our pervasive middleware development. What distinguishes our work from other types of middleware offering similar functionality is that we targeted the middleware for mobile devices, avoiding technologies that would hinder its usage in such resource-limited devices.

Components

Our middleware architecture divides functionality into components, each of which specializes in some task or domain area and provides corresponding functionality to other system entities. This CAPNET (Context-Aware Pervasive Networking) middleware masks the complexity of networks and distributed systems, letting developers focus on application-specific issues (see figure 2). Furthermore, it places the commonly used functions in independent components, so they can be shared across platforms and software environments.

A component can represent a service in the user's environment, such as a printer or a projector. The components live inside a container called an *engine*. The engine core fulfills the five fundamental requirements. It handles interoperability with the other engines and components in the system. It provides location transparency to the components and mechanisms for discovering resources, services, and other system components. Context awareness is provided to the system by context components, and dynamically reconfigurable stub objects support adaptability. The engine can also contain an unspecified number of other components that add value to the system.

The applications (that is, the services the user interacts with) are also components. They can use other components in the system, regardless of the application's or components' physical location. Figure 3 illustrates a system consisting of a mobile phone and two network servers. The application, running on the mobile phone, uses keyboard and projector services, represented by corresponding components.

Engine core

The engine core consists of service discovery, messaging, context, and component management components. Component management is an engine's main controlling unit. It controls all components by loading them into the system and initializing, starting, and terminating them. It also processes requests from the other components concerning component access. The messaging component provides synchronous and asynchronous communication mechanisms to other components. Furthermore, messaging provides distribution and interoperability mechanisms between the components. All communication between components in different engines flows through the messaging component.

The service discovery component discovers and locates services, resources, and other components in the environment. When one component (client) needs another's service, it selects one from a list of alternatives provided by the service discovery and asks component management to provide a reference to the selected component. The context component provides context information, both synchronously and asynchronously, as context events. It uses the available sensors to acquire contextual data from the environment.

Dynamically reconfigurable stub objects provide location transparency to the CAPNET system. When an application requests a component, component management returns a stub object, representing the requested component. If the requested component is outside the local engine, the stub object uses messaging to redirect component calls to the actual component in a remote engine. If an application's execution environment changes owing to the user's mobility, component management can adapt the system to the new situation by reconfiguring the stub objects and associating them with the corresponding services in the new environment.

Design

Our framework design started from this middleware. We aimed to fulfill the fundamental requirements for generic-use pervasive middleware. We tested the middleware with several prototypes before introducing the RFID tags into the framework. Because the system interprets data read from tags as contextual information, adding the RFID tags into the framework was straightforward. The middleware generates context events when users touch RFID tags. The applications react to these events either by offering the requested service themselves or by requesting from the middleware the component offering the service and redirecting the service request to that component. In designing the visual symbols and the data representation, we wanted to produce generic tags not tied

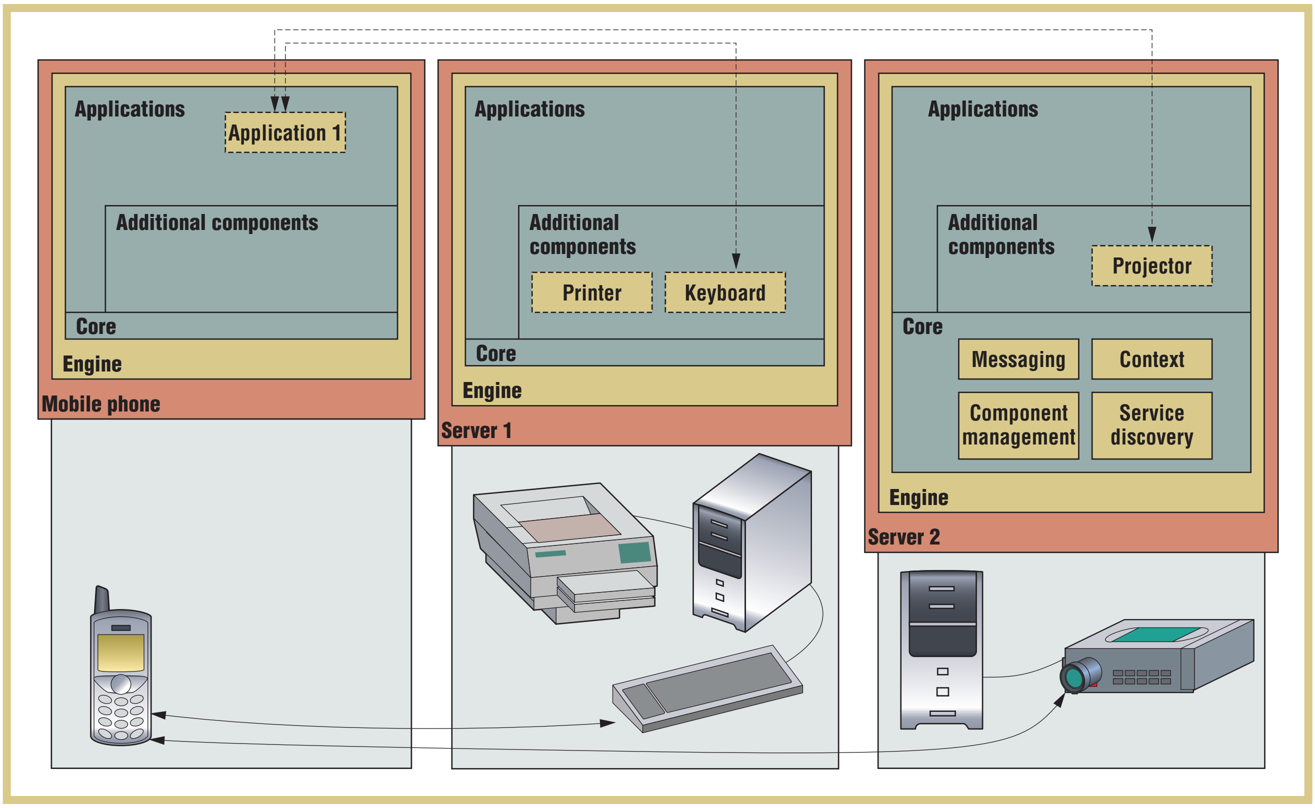


Figure 3. Capnet engines and services.

to any specific application, which you could use to request services from a wide set of applications.

Usability and user experience tests

Our first usability tests used a prototype that consisted of applications running on a mobile phone and several services in the environment. The middleware for the Symbian mobile phones included component management, messaging, and context components. We implemented complete middleware on network servers in Java. We equipped the mobile phone with an external RFID reader module that communicated with the mobile phone via Bluetooth. We used the visual tags in figure 1 to label the services. We hadn't yet fully implemented the data representation, but the middleware hid this deficiency.

The test aimed to study how well users could understand the visual symbols' meaning. We also wanted to compare the general and special tags. Moreover, we were interested in the user experience, acceptability, and first impressions of this new technology. The test's participants consisted of university students and staff. Eight individuals formed four pairs. The average age was 28 and all of the participants were men. The participants evaluated themselves as having good skills in using this kind of technology. Almost everyone was familiar with the phone model used in the tests (Nokia 6600). Also, technology played an important part in their everyday lives. The participants were interested in using new technology and had a positive attitude toward new technology in general. In spite of that, they initially had reservations about using RFID tags. We viewed the selected user group as potential early adopters. According to Everett Rogers, early adopters provide opinion leadership to others and are the first in their group to adopt, maintain, and evaluate innovations for the others.[9]

Scenario

We collected the material for this study using thinking-aloud methods—the participants continuously thought out loud while carrying out the given tasks. We also used observations, forms, and discussions as tools. We observed the teams carry out tasks over two days. After they finished a task, we asked the participants to fill out question forms. In these tests, touching a general symbol always triggered a service that listed on the terminal's display the other services available for this object. The second alternative was to touch a special symbol that offered direct access to a service. In the tests, the participants played Kari's role in the following scenario:

Kari is a new employee in a company and has been invited to a meeting with his manager Janne. However, Kari finds out that Janne isn't in his office at the appointed time. Kari has been told that the building is equipped with services that people can access by touching tags with

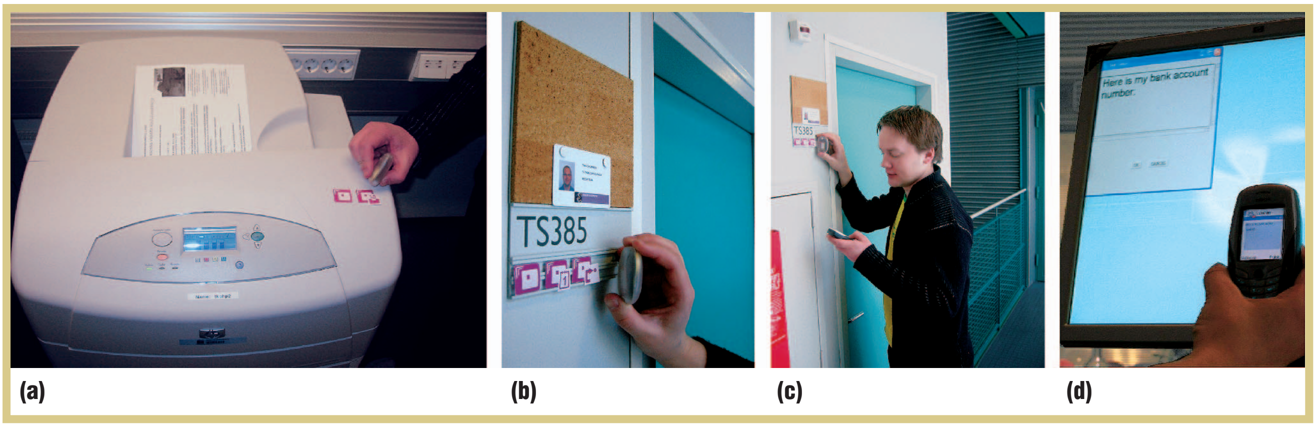


Figure 4. User test scenario: (a) using the printer, (b) locating another user, (c) accessing the calendar, and (d) transferring the SMS application interface to the computer screen.

a mobile phone. The tags have easily recognizable visual symbols that suggest the offered service. Kari notices some tags next to the door of Janne's office, one of which looks like a calendar. Intuitively, Kari associates the calendar tag with Janne's schedule and touches it with his phone. This action opens a calendar application with Janne's calendar appointments on Kari's phone display. It appears that Janne has scheduled the meeting in a meeting room instead of his office.

Kari notices a map tag next to the calendar tag. As Kari touches it with his phone, the phone opens an application that shows a route to Janne's location, which is the meeting room. As Kari walks toward the room, he notices a printer in the corridor. Kari finds out that the printer is labeled with two tags; the first one is a general tag. Kari associates the second tag with a printing action and touches it with his phone. Because the map application is on top on the phone's display, it prints the map. Out of curiosity, Kari also touches the first tag. This causes his phone to show a list of available services: Print, Info, and Call maintenance. So, Kari finds out that, with a general tag, he can access all of the printer's available services. Finally, Kari finds the meeting room. As he's about to enter it, he discovers some tags next to the door. He recognizes one tag as the meeting profile and touches it with his phone. Consequently, the phone changes the profile into the meeting mode, and no phone calls disturb him during the meeting.

When the meeting is over, Kari needs to make a reservation for a cocktail party in the evening. He finds some tags in the brochure of the company organizing the party. A tag is next to a contact number. Kari associates this tag with a phone call and touches it. A phone call is initialized, and Kari makes the reservation. Finally, before going back home, Kari finds out he has to enroll in a weekend course. While he's writing a SMS (short message service) message to the course assistant, he notices a computer with a tag attached to it. Touching the tag enhances the mobile phone with advanced I/O capabilities: the SMS application's display is transferred to the computer's larger screen, letting him type the text using the computer's keyboard.

Figure 4 illustrates some stages of the test. In the pictures, the user is holding the tag reader that communicates via Bluetooth with the mobile phone. In figure 4a, the user activates the printer service. Next (see figure 4b), the user locates another employee in the building. In figure 4c, the user is at the office doorway and uses the services he has found. In figure 4d, the SMS application's user interface has been transferred onto the computer's screen.

Results

We roughly categorize the results under four topics: usability, control, security, and utility and social acceptability.

Usability. The users were able to learn how to use the special tags during their first usage situation. Users preferred special tags over general tags because the special tags present their meaning in a concrete way that they felt was logical, fast, and easy to use. However, the users noticed that finding the correct symbol for a certain task among several adjacent special tags could be difficult. The users also appreciated the simplicity of having a single general tag. The users valued the colors and simple symbols as well. They wanted symbol features that were familiar from earlier usage contexts; clearly, familiar features facilitate recognition in new usage situations. Finally, symbol placement was found important in terms of usability. Consistent placement helps in recognizing and interpreting symbols.

Related user trials of selecting mobile services indicate that physical pointing can be significantly faster than the conventional method.[10] Usability studies concerning a mobile device's menu structures suggest that novice users like hierarchical menus while experts might wish for shortcut codes that let them directly access the desired task.[11] Such results, together with our observations of a fast learning rate, indicate that the special tags suit expert users—if we provide unambiguous visual symbols. Users' *perceived affordance*[12]—or perception of what a visual symbol stands for or activates—must match the reality. Lars Erik Holmquist and his colleagues also emphasize the importance of affordance.[13] Further complicating this is that users can give different meanings to a symbol depending on the usage situation and their personal background. For example, the test users noted that they interpret an envelope symbol on a computer display as a symbol for email but on a mobile phone display as a symbol for SMS.

Control. During the tests, we also discussed automated service activation—services that could activate when a user enters the tag's proximity. The users preferred clear manual user interaction because it gives a better feeling of control. Related to this was the finding that feeling in control often requires providing the user with clear feedback, although feedback wasn't required if the service started immediately after touching a tag. Other researchers also mention feedback as a key usability issue.[10] Although users preferred special tags, general tags produced a better feeling of control because they could select services from a menu.

Security. Users found special tags more secure than general tags because they give more information about the expected system behavior. The mobile phone was important for everyone; users mentioned it as a natural, familiar, and secure terminal for the tested services. However, to assess security, the users wanted to know who was behind the services. They didn't want to risk their personal mobile phones if they doubted a service's security. They felt that new technology was secure when used in a familiar environment—but that touching tags with their own mobile phone in public places was risky. Eija Kaasinen discovered similar results regarding control and security.[14]

Utility and social acceptability. Users evaluated the services' utility on the basis of their personal needs. They found the tested services to be useful. The special symbols resulted in higher utility because they were faster and easier to use. However, utility alone doesn't suffice for suggesting implementation in daily use—acceptance is also required. Users felt the mobile phone was an acceptable terminal for the tested services. The test users were also ready to use these services immediately or within one year in their home environment if the required facilities and a positive atmosphere existed. They weren't ready to use the services elsewhere because they considered the security risk too high—and because they were embarrassed about touching RFID tags in public places. For example, they felt that touching might seem too eye-catching on the street or at a shopping mall. So, although touching is itself a natural action, it isn't necessarily considered a socially acceptable way to use new technology.

We found social acceptability and security to be key factors in deploying RFID tags in our everyday environment. The environment itself impacts this feeling of security.

A user's feeling of control seems to considerably influence the adoption of new technologies such as RFID tags. And while designing unambiguous symbols is a challenge, such symbols (perhaps well-known local symbols) in the user interface can help promote acceptance of new technology. You should limit special symbols to a carefully selected set of common services. This prevents the sheer number of symbols from confusing or delaying the user's selection of a service. It suffices to have other services accessible by first touching a general tag and then selecting a service from the list on the terminal's display.

We also found social acceptability and security to be key factors in deploying RFID tags in our everyday environment. The environment itself (private versus public) impacts this feeling of security. Acting under the watchful eyes of strangers or in an otherwise negative atmosphere might be a barrier for some users. These findings suggest that the first RFID applications should be for familiar environments (for example, homes)—users would trust the applications and wouldn't be ashamed of touching tags with a mobile phone. After users got accustomed to RFID tags, public use might have a lower threshold. Alternative physical selection methods might also promote wider usage; for example, some users might prefer to request services by pointing to visual symbols instead of touching them.

Our findings fall in line with other researchers' results. However, the variability in usage situations inherent in pervasive applications introduces new challenges that need further research before we can guarantee good usability and a positive user experience. The tests we reported are quite modest and, hence, preliminary. We'll continue our research and continue to further develop the middleware. We have RFID readers integrated into mobile phones, so we can start studying how users regard touching tags with a phone instead of a separate reader. We are keen on finding new tools for designing visual symbols through the concept of perceived affordance and are creating a wider set of visual symbols with the objective of producing a set of tags that many different applications can use. We plan on conducting more usability and user experience studies to verify and further elaborate the findings with a greater number and diversity of users. We hope to also clarify the factors affecting acceptability and interpretation in different environments and contexts. ▣

ACKNOWLEDGMENTS

The National Technology Agency of Finland funded this work. We thank all the personnel in the CAPNET program and the participating companies. We also acknowledge Simo Hosio, Marketta Heinonen, and Heikki Laaksamo for creating the test environment, designing the visual tags, and specifying the EPC representation, respectively.

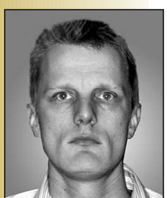
REFERENCES

1. B. Ullmer and H. Ishii, "Emerging Frameworks for Tangible User Interfaces," *Human-Computer Interaction in the New Millennium*, J.M. Carroll, ed., Addison-Wesley, 2001, pp. 579–601.
2. J. Rekimoto, B. Ullmer, and H. Oba, "DataTiles: A Modular Platform for Mixed Physical and Graphical Interactions," *Proc. SIGCHI Conf. Human Factors in Computing Systems*, ACM Press, 2001, pp. 269–276.
3. R. Want et al., "Bridging Physical and Virtual Worlds with Electronic Tags," *Proc. SIGCHI Conf. Human Factors in Computing Systems: The CHI Is the Limit*, ACM Press, 1999, pp. 370–377.
4. C. Swindells et al., "That One There! Pointing to Establish Device Identity," *Proc. Symp. User Interface Software and Technology* (UIST 02), ACM Press, 2002, pp. 151–160.
5. H. Ailisto et al., "A Physical Selection Paradigm for Ubiquitous Computing," *Ambient Intelligence*, G. Goos et al., eds., LNCS 2875, Springer, 2003, pp. 372–383.
6. M. Roman et al., "A Middleware Infrastructure for Active Spaces," *IEEE Pervasive Computing*, vol. 1, no. 4, 2002, pp. 74–83.
7. R. Grimm, "One.world: Experiences with a Pervasive Computing Architecture," *IEEE Pervasive Computing*, vol. 3, no. 3, 2004, pp. 22–30.
8. D. Garlan et al., "Project Aura: Toward Distraction-Free Pervasive Computing," *IEEE Pervasive Computing*, vol. 1, no. 2, 2002, pp. 22–31.
9. E.M. Rogers, *The Diffusion of Innovations*, 4th ed., Free Press, 1995, p. 518.
10. L. Pohjanheimo, H. Ailisto, and J. Plomp, "User Experiment with Physical Pointing for Accessing Services with a Mobile Device," *Proc. European Symp. Ambient Intelligence Workshop on Ambient Intelligent Technologies for Wellbeing at Home* (EUSAI 04), 2004, www.eusai.net/MyReview/FILES/WORKSHOP/7369a7036386b928a89ce747bddd4395.pdf.
11. K. Hassanein and M. Head, "Ubiquitous Usability: Exploring Mobile Interfaces within the Context of a Theoretical Model," *Proc. 15th Conf. Advanced Information Systems Eng.* (CAiSE 2003), 2003, http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS//Vol-75/files/UMICS_16.pdf.
12. D.A. Norman, "Affordance, Conventions, and Design," *Interactions*, ACM Press, 1999, vol. 6, no. 3, pp. 38–43.
13. L. Holmquist, J. Redström, and P. Ljungstrand, "Token-Based Access to Digital Information," *Proc. 1st Int'l Symp. Handheld and Ubiquitous Computing*, LNCS 1707, Springer, 1999, pp. 234–245.
14. E. Kaasinen, "User Acceptance of Mobile Services—Value, Ease of Use, Trust and Ease of Adoption," doctoral dissertation, VTT 566, VTT Publications, Espoo, Finland, 2005; www.vtt.fi/inf/pdf/publications/2005/P566.pdf.

the AUTHORS

Jukka Riekki is a professor of software architectures for embedded systems in the University of Oulu's Department of Electrical and Information Engineering. His main research interest is context-aware systems serving people in their everyday environment. He received his Doctor of Technology degree from the University of Oulu. He's a member of the IEEE. Contact him at the Dept. of Electrical and Information Eng., PO Box 4500, FIN-90014 Univ. of Oulu, Finland; jukka.riekki@ee.oulu.fi.

Timo Salminen is a PhD student in the University of Oulu's Department of Electrical and Information Engineering. His research interests include pervasive computing, mobile computing, middleware, and networking. He received his MSc in computer engineering from the University of Oulu. He's a member of the IEEE. Contact him at the Dept. of Electrical and Information Eng., PO Box 4500, FIN-90014 Univ. of Oulu, Finland; timo.salminen@oulu.fi.



Ismo Alakärppä is doing doctoral research at the University of Lapland on the technology acceptance of the elderly. His research interests include an ecological approach to design and technology acceptance and its relation with industrial design. He received his MA in industrial design from the University of Lapland. He's a member of the International Society for Gerontechnology. Contact him at the Dept. of Industrial Design, PO Box 122, FIN-96101 Univ. of Lapland, Finland; ismo.alakarppa@ulapland.fi.