

Interaction in Pervasive Computing Settings using Bluetooth-enabled Active Tags and Passive RFID Technology together with Mobile Phones

Frank Siegemund and Christian Flörkemeier
 Institute for Pervasive Computing
 Department of Computer Science
 ETH Zurich, Switzerland
 {siegemun|floerkem}@inf.ethz.ch

Abstract—Passive RFID technology and unobtrusive Bluetooth-enabled active tags are means to augment products and everyday objects with information technology invisible to human users. This paper analyzes general interaction patterns in such pervasive computing settings where information about the user’s context is derived by a combination of active and passive tags present in the user’s environment. We distinguish between interaction that is initiated by smart objects and interaction initiated by human users, and investigate how an association between interaction partners can take place through explicit human actions as well as invisibly through context information provided by active tags. The concept of invisible preselection of interaction partners based on the user’s context is introduced. It enables unobtrusive interaction with smart objects in that it combines different forms of association, e.g. implicit and user initiated association, by transferring interaction stubs to mobile devices based on the user’s current situation. Invisible preselection can also be used for remote interaction. By assigning phone numbers to smart objects, we propose making this remote user interaction with everyday items and their virtual counterparts as easy as making a phone call. Mobile phones are also used as mobile infrastructure access points for smart objects. We evaluate the suitability of the proposed concepts on the basis of three concrete examples: a product monitoring system, a smart medicine cabinet, and a remote interaction application.

I. INTRODUCTION

Pervasive computing envisions a world of omnipresent but invisible information technology embedded into products and everyday items [16] [27]. In this paper, we investigate interaction patterns in environments where passive RFID labels and active Bluetooth-enabled tags are attached to products and everyday objects. The tags are so unobtrusive that they do not change the physical appearance of objects and are ideally invisible to users. Furthermore, interaction with them does not require explicit human actions to be initiated – as opposed to barcodes that require users to explicitly scan codes, or infrared devices that often need manual alignment to ensure line-of-sight for communication. Interaction can be initiated by smart objects as well as by users, and communication with users’ mobile devices can take place either with the user being aware of

it or alternatively unnoticed by the users. The fact that interaction can be initiated without explicit human intervention is also a key concept in successful industrial applications of active and passive tags, such as in supply chain management [8] [10].

By using the information stored on RFID tags and the ability of active tags to sense their environment through sensors, to carry out computations and to communicate with peers, the context of users and the state of smart objects can be determined collaboratively. This information can be used to influence the way an interaction takes place, and it can be used to decide what entities are available to communicate. We investigate different forms of association between interaction partners that take place through explicit human actions as well as through side effects of users’ normal behavior, i.e. invisibly for them. Furthermore, we combine these different approaches by introducing the concept of invisible preselection of interaction partners based on the user’s context. It enables unobtrusive interaction with smart objects taking place at different locations and at different times.

The applications described in this paper make use of Bluetooth-enabled active tags – also referred to as BTnodes [3] [4]. The BTnodes were partially developed within the Smart-Its project [23]. The main reason for using Bluetooth as communication standard for the active tags is that Bluetooth modules are being integrated in an increasing number of consumer devices such as mobile phones, PDAs, and digital cameras. Therefore, the Bluetooth standard serves as a bridge to users’ mobile devices and as an easy means to interact with people. In this paper, mobile phones serve as the major platform for users to communicate with smart objects. According to Forrester Research [11], the mobile phone market in some European countries is beginning to reach its saturation with coverage rates of about 80%. Mobile phones are a technology that has become almost ubiquitous. Because mobile phones are carried around by their users, they are also present when an interaction with a smart object is to take place. Furthermore, mobile phones offer many possibilities for smart objects to interact with people by using features that are familiar to the vast majority of mobile phone users such as short message service messages (SMS), alarms, custom calls, or OBEX objects like calendar entries and business cards. The fact that 78.7 billion SMS messages were sent

in Western Europe in the year 2001 [15] shows how popular those features are. We try to use these well established interaction patterns in connection with smart objects.

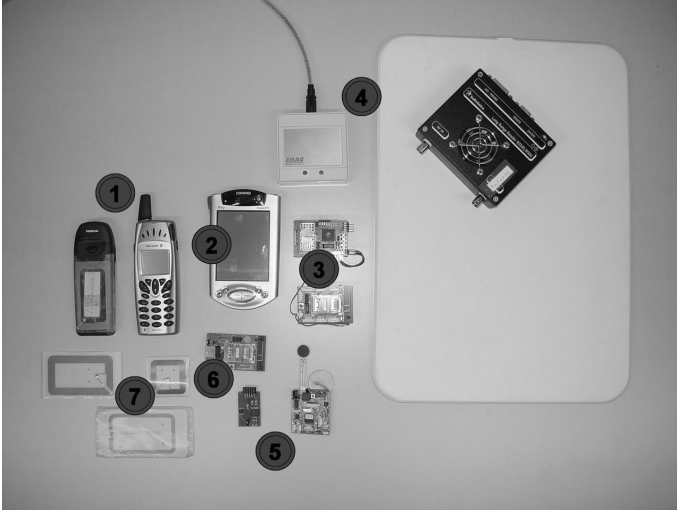


Fig. 1. Some of the devices used to evaluate interaction patterns with smart objects: Bluetooth-enabled phones tagged with RFID labels (1), PDAs (2), BTnodes (3), RFID antennas and readers (4), sensor boards (the one on the right was developed by TecO, University of Karlsruhe) (5), Bluetooth access points (developed by TIK, ETH Zurich) (6) and RFID tags (7).

Passive RFID tags become increasingly important in business processes and are likely to become as ubiquitous as barcodes. Initiatives such as the Auto-ID center at MIT [1], which works on low cost tags and a networking infrastructure to support them, will help speed up the adoption of pervasive RFID tagging. By attaching RFID scanners to BTnodes, we can bridge the gap between active and passive tags and use both techniques for implementing interaction patterns in smart environments. Data stored on an RFID tag (e.g. an electronic product code) often cannot be semantically interpreted by a small peer-to-peer network that exists between active tags in the user's environment but requires access to a background infrastructure. In our approach, Bluetooth-enabled mobile phones are used as mobile access points for smart tags, allowing them to access background infrastructure services and virtual counterparts when a user is in range of a smart object.

The remainder of this paper is structured as follows: Section II analyzes common interaction patterns with smart objects equipped with active or passive tags and motivates the concept of invisible preselection of interaction partners based on the user's context. Section III introduces three different scenarios that show how the different forms of interaction emerge in concrete applications, how people can interact with smart devices independent from their current location, and how hybrid approaches for the association of interaction partners can improve interaction in the envisioned environments. Section IV describes the technical realization of the scenarios and evaluates the core components of the underlying architecture. Section V gives an overview on related work. Section VI concludes the paper.

II. INTERACTION WITH SMART OBJECTS

Communication in pervasive computing settings occurs between smart objects, between smart objects and background infrastructure services, and between smart objects and their users. Context information derived collaboratively by active tags attached to the objects can improve all those different kinds of interaction considerably [21] [22]. In this section we analyze different forms of interaction between human users and smart objects and argue in favor of hybrid approaches for the association of interaction partners in the envisioned settings.

A. Classification of Interaction Patterns

Active tags and passive RFID labels do not possess screens or provide additional buttons, keyboards, or any other means for users to physically interact with smart objects. How do people communicate with augmented objects although they cannot see the tags and might not even know which objects are smart? BTnodes and RFID tags are ideally invisible to users and unobtrusive to such a degree that they do not disturb the way in which people do normally use their items. Intelligent tags should merely add additional functionality to an object without disturbing the way people usually interact with them.

In the following, we distinguish between interactions initiated by users and interactions initiated by smart objects (cf. figure 2). In the former case, users have the intention to interact with an object and must therefore have the possibility to address this object. In the latter case the smart objects, that is, the active tags attached to them, trigger an interaction because of certain state changes in their environment and must be able to identify persons with whom an interaction is to take place.

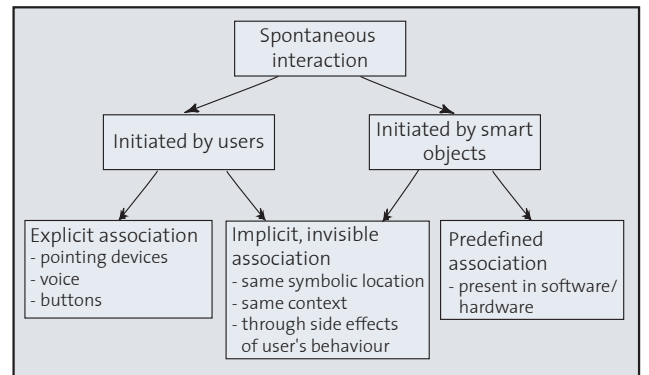


Fig. 2. Association between interaction partners in pervasive computing settings

When interaction is initiated by human users, there are basically two alternatives to associate the user with a certain smart object: explicit and implicit association. The first option uses explicit actions a person would not be using under normal circumstances to address an object, for example by speaking to an item or by using a laser pointer to select an object [18]. The main advantage of using explicit actions is that the user is in full control of the association process. On the other side, he/she must be aware of which objects are augmented and must be familiar with the new method to interact with them.

Alternatively, implicit association of interaction partners can take place as a side effect of users' normal behavior while handling an object. Here, existing interaction patterns in connection with the object or product in its unaugmented form are used to establish an association, i.e. the association takes place invisibly for the user. This approach requires that a smart object can sense when it is used or going to be used (for example an automatic door senses when someone approaches it and knows that it is going to be used). This can be done by considering sensory input and derived context information of the object itself and in collaboration with nearby objects. Examples for these kinds of context information are: a person is approaching an automatic door, a specific kind of medicine was taken out of a medicine cabinet, a movement in the range of an automatic light switch, two persons are running together, a specific person shared the same symbolic location with an object for a dozen of times. This kind of association that is based on such context information is called invisible, implicit association. Here, a user is not forced to learn additional interaction patterns and does not even need to know which objects are augmented. The main disadvantage of this approach is that the user is not in full control of the association process. Therefore, it is only useful when simple sensory input and the very restricted computational capabilities of active tags are sufficient to anticipate an interaction with high probability, which is seldom the case.

When interaction is initiated by smart objects, there also exist two possibilities to associate interaction partners: predefined and implicit association. The first alternative is to store the address of interaction partners, e.g. the GSM number or Bluetooth device address of a mobile phone belonging to a specific person on the device. Considering for example a simple notification service, the smart object would have predefined rules about whom to contact in what situation, which is inflexible. Changes require reprogramming or even restructuring of hardware features.

As with interaction initiated by users, another possibility to determine interaction partners is to use sensory input and derived context information of augmented objects. Considering again a notification service, a smart object would try to find persons that share a certain symbolic location with the object, e.g. people that are in the same room, and would then notify these persons. Again, this approach requires that a smart object can perceive its environment and collaborate with other objects in its proximity. In section III we illustrate this with a concrete application.

B. Invisible Preselection of Communication Partners Based on the User's Context

The three forms of association shown in figure 2 - explicit, invisible, and predefined association - in their pure form are often unsuitable in pervasive computing settings. For example, when there are huge numbers of potential interaction partners in range, or in environments where people do not know which objects are augmented, direct manipulation of interaction partners becomes difficult and therefore explicit association almost impossible. On the other hand, pure implicit association has the drawback that smart objects might initiate interactions unwanted by users, which is intolerable when users are being con-

tinuously disturbed by requests from their "smart" objects. We therefore suggest to use a hybrid approach for association in pervasive computing environments that reduces the number of potential interaction partners through implicit association but still leaves it to the user to explicitly establish an interaction with preselected objects. Preselection takes place completely unnoticed by users, who are therefore not disturbed by this process. Also, if the set of preselected devices does not contain the one expected, it does not exclude the possibility of additional explicit association. Moreover, there are certain kinds of interactions that take place at different points in time and at different locations. Invisible preselection is also an approach to enable these kinds of interactions.

Sensory input and derived context data as well as history information collected by and exchanged between smart objects in the user's environment is used to exclude entities for interaction. Exclusion of interaction partners takes place invisibly for the user, who can use explicit actions afterwards to initiate an interaction with preselected devices. By using this approach the number of potential interaction partners is decreased making an explicit association easier. Figure 3 depicts the core concept of invisible preselection: implicit association is used to reduce the number of potential communication partners. During this selective process, objects are selected for later interaction that might take place at another location. In order to make later interaction possible, information about how to interact with the selected objects is stored on a user's mobile device, e.g. a phone book entry for a smart object on a mobile phone. This instance that makes later interaction possible is called an *interaction stub*. Interaction stubs are stored on users' personal devices that are carried around by the users. Later, possibly at a different location, a person chooses one of the stubs through a conscious, explicit action and initiates an interaction with the smart object.

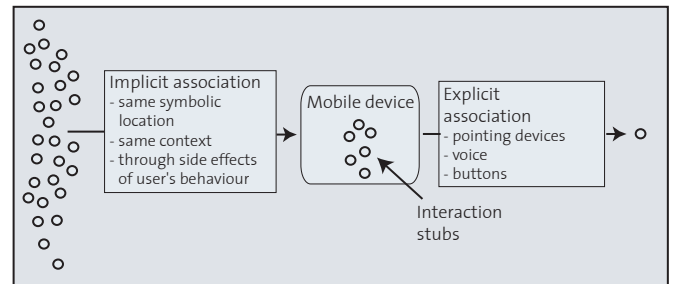


Fig. 3. Invisible preselection of interaction partners

Invisible preselection assures that interactions that are very unlikely to happen are hidden from the user. Because there are many interactions possible (potentially with each object in the user's proximity and other objects that can be controlled remotely) invisible preselection assures that users are not overloaded by unwanted interaction requests from smart objects. Invisible preselection makes a comfortable and convenient form of interaction in pervasive computing settings possible. When an interaction is initiated by a smart object and not by implicit or explicit actions of users, the preselection can also be carried out by another smart object. The first object can then choose an interaction partner based on its sensory input from the set of preselected entities.

III. SCENARIOS

The three sample applications described in this section illustrate the different interaction patterns discussed previously. All scenarios have in common that they incorporate active and passive tags that are attached to products and everyday items. When developing these applications, we have focused on augmenting those everyday items in such a way that the user does not have to change the way he/she interacts with them in his/her daily life. Section IV provides a more detailed analysis of the technical components on which the following applications rely.

A. Smart Product Monitoring

The smart product monitoring scenario is an example for a pervasive computing scenario where interaction is initiated by a smart object. The object chosen for this prototype is an egg carton representing an arbitrary fragile object that is in storage e.g. in a warehouse. The object is augmented in such a way that it detects whenever it is dropped or not stored within the appropriate temperature range. Whenever such an exception occurs, it triggers an alarm by informing the appropriate contact person via an SMS.

The challenges when realizing this application were to monitor the physical object unobtrusively (1) and to associate the appropriate contact person with the smart object without explicit manual pre-configuration (2).



Fig. 4. An egg carton augmented with a Bluetooth-enabled active tag.

To monitor the egg carton, BTnodes with a sensor board for acceleration and temperature sensors are attached to the fragile product (see Figure 4). Based on sensory input from acceleration and temperature sensors, the state of the object is determined. If the egg carton falls down or is kept for too long under unsuitable conditions, the BTnode activates its communication module to send an alarm. The BTnode sends the notification embedded in an SMS message to the appropriate contact person via a Bluetooth access point that offers a gateway to the cellular phone network. The Bluetooth access point then forwards the message to a mobile phone. As such the scenario is an example of a context-triggered action [19], where the sensing of a context change (e.g. broken eggs) triggers the alarm to the user.

The SMS message sent to the user contains not only a short description about the current state of the product, but also a range of commands that can be sent back and processed by the smart product. The user replies to the incoming message by activating embedded commands and sends a resulting SMS message back to the smart object. It carries out these commands and sends another message containing the results back. Consecutive messages can be exchanged between interaction partners. For the user, the contact information of a smart product, i.e. its phone number, is implicitly given, because it sent the first message. Figure 5 shows an example of this process.



Fig. 5. An SMS notification received from a smart product by a mobile phone (1), a response message with activated history command (2), and the corresponding result from the egg box (3).

The question how the smart object knows to what mobile phone number it needs to send the notification without explicit, manual pre-configuration is solved using passive RFID tag technology. The mobile phones of the potential contact persons are equipped with RFID tags. Whenever they enter the neighborhood of the smart object (for example a lorry or a certain room the product is in), a wireless RFID reader attached to a BTnode or a Bluetooth-enabled PDA communicates the presence of the tag together with service parameters, which describe how to access the device (e.g. its GSM phone number and Bluetooth address) via Bluetooth broadcast to the active tag in the egg carton.¹

Using the information from the RFID reader, the active tag in the egg carton is now aware of who is sharing a symbolic location with the object it is augmenting and for how long. This location context and the history of that information allows the active tag to generate an interaction stub that contains the appropriate contact information. This interaction stub is executed by the BTnode once an alarm needs to be triggered.

The infrastructure described above for the invisible preselection of interaction partners can also be reused to make the appropriate choices at the moment the alarm is triggered. If the previously selected interaction partner can be reached at that specific instance via more than one communication medium

¹Bluetooth only supports a piconet broadcast; see section IV for a detailed description of the technical challenges.

(e.g. GSM and Bluetooth), the active tag can determine the most appropriate one.

B. Remote Interaction with Smart Objects and Locations

In the second scenario, the users initiate the interaction by remotely querying smart physical or logical objects using their mobile phones. It exemplifies the concept of invisible preselection from a user perspective.

In order to make remote interaction possible, each augmented everyday object is assigned a telephone number. By using this telephone number, a user can simply call a certain object and interact with it. In our application, we use SMS messages to implement this interaction, because virtually every mobile phone user is familiar with this phone feature, and because text based messages can be easily processed by resource-restricted active tags such as the BTnodes. Messages sent to the smart objects via the cellular phone network contain commands that are processed by their active tags.

However, people usually are not willing to memorize the phone numbers and all different forms of commands that can be sent to a smart object. We therefore implemented an approach where phone book entries for everyday items and SMS templates containing the commands supported by them are sent to user's mobile phones based on context information derived by active tags in the user's environment. SMS templates and phone book entries are the *interaction stubs* that enable remote interaction independent from the user's current location. The smart objects that have such an interaction stub on the user's mobile phone are chosen in an implicit preselection phase.

Activation and modification of commands in SMS messages have to be as simple as possible. In the approach we have taken, all valid commands are embedded in an SMS template and in their simplest form can be activated by deleting just a single character. This requires only minimum effort from the user (cf. figure 6).



Fig. 6. Phone book entries for preselected smart objects (1), a list of corresponding SMS templates (2), an edited SMS template with activated command (3), and the response message (4).

In order to store the interaction stubs of smart objects in the user's mobile phone, he/she once must have been in radio range

of a BTnode attached to this object. However, because there are so many potential communication partners, not every smart object can send interaction stubs to the mobile device, and an implicit preselection phase becomes necessary. This preselection is done invisibly for the user on the basis of sensory input and history information of active tags.

In the following this is described by means of an augmented logical object: an office. Information about how often a person shares a certain symbolic location with a BTnode (i.e. how often a specific person is in the room) and how long he/she stays there is used to decide whether an interaction stub is downloaded to the mobile device. When people enter the office they are identified via RFID tags integrated into their wallets or attached to their mobile devices. The information on the corresponding RFID label is sent to all smart objects in the office. Smart objects know to what room they belong, because they are also equipped with RFID tags. When they enter the room this information is also broadcasted to them. By using the described approach, an object can decide what persons share a certain symbolic location with it rather than who is in radio range of the object. This is a significant advantage, since mobile devices of people in another room or adjacent building that may have still radio connectivity to the smart objects in the office usually do not want to interact with them.

C. The Smart Medicine Cabinet

The smart medicine cabinet is an application that was designed to support mobile and young patients with chronic diseases. It is supposed to improve the drug compliance of these patients by reminding them to take their medicine. The smart medicine cabinet also knows about its contents so that the user can query it remotely to check which medication he/she has currently available. Other features include out-of-date detection and alarms for potential product recalls. The main requirement when realizing the above services was to avoid changing the routine with which the user typically goes about taking his/her medicine, e.g. by requiring him to manually scan the barcode on the medication or configuring software on a personal computer.

To realize the above scenario and requirements the following technologies were incorporated into an ordinary medicine cabinet as shown in Figure 7: (1) passive RFID tags on the folding boxes combined with a medicine cabinet, which was equipped with an RFID reader; (2) an active tag that processes the information from the RFID reader and communicates via Bluetooth with a (3) mobile phone.

This application distinguishes itself from the previous scenarios for two reasons: (1) it uses passive RFID technology for the actual monitoring of the physical objects (the drugs in this case) and (2) the mobile phone is not only used as the user interface, remote communication link and storage medium for interaction stubs, but also as the local access point for the augmented objects. This results in the fact that the medicine cabinet operates in a disconnected mode whenever there is no mobile phone present. It hence requires a virtual counterpart in the background infrastructure that represents the medicine cabinet continuously. Whenever a mobile phone is in the vicinity of medicine cabinet and provides connectivity, the BTnode in

the medicine cabinet synchronizes with this virtual counterpart. Remote queries similar to the ones described in the previous section can now no longer address the smart object, but need to address its virtual counterpart.

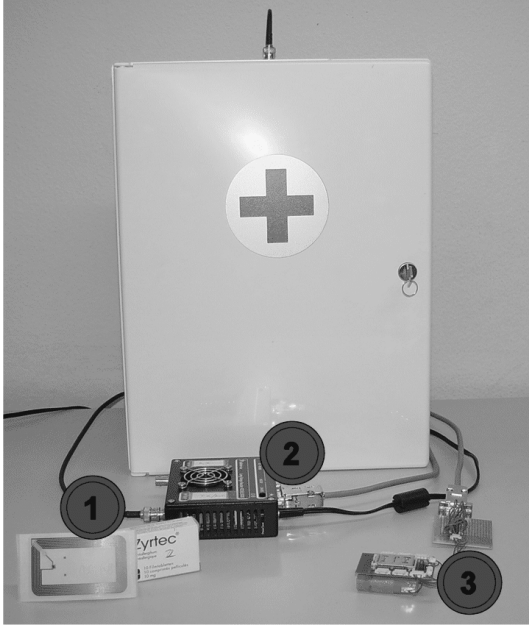


Fig. 7. The medicine cabinet together with the information technology that augments it: RFID tag (1) and reader (2), Bluetooth node (3).

The application keeps track of the medicine in the cabinet by reading the serial number on the RFID tags with which the folding boxes have been equipped. It uses this serial number as the global key to look up information specific to the type of product as well as information specific to this instance of medication such as the user's prescription information and the expiry date. The communication link to the backend infrastructure where these data are stored is established whenever the user is in the vicinity of the medicine cabinet with his/her Bluetooth equipped mobile phone. Drug usage is monitored by monitoring the regular "appearance" and "disappearance" of an RFID tag as the patient removes the medication from the cabinet and from the read range. This usage information is also sent via the mobile phone to the virtual counterpart of the folding box in a backend infrastructure.

The application reminds the patient to take his/her medicine by programming alarms into the mobile phone according to the prescription information associated with a certain type of medication in the medicine cabinet (see Figure 8 for details). The prescription information is initially stored with the virtual counterpart and is transmitted to the medicine cabinet during the next synchronization. The alarms are transferred to the mobile phone by using the Bluetooth link between the active tag that "manages" the medicine cabinet and the mobile phone.

The main motivation for using passive RFID technology is that it allows for monitoring the inventory without the user "noticing" or even having to "help". In combination with the active tag that manages the RFID reader and the connection to the mobile phone that means that there is "zero configuration or interference" required from the user.



Fig. 8. An alarm previously written to a mobile phone from the medicine cabinet to remind patients to take the medicine Zyrtec (1, 2), and an SMS template that can be used by a patient to get further information about this medicine by asking the virtual counterpart (3).

IV. TECHNICAL REALIZATION

This section describes how the different interaction patterns motivated in section III were realized. The remote interaction and smart product monitoring scenario are based on the same architecture, which is described in subsection IV-B. Here, interaction, even if it is done remotely, takes place directly with the smart object. When a person calls a smart object, an interaction request reaches and is processed by the smart object itself. However, in the smart medicine cabinet scenario, remote queries are processed by a virtual counterpart, which is updated each time a smart object can connect to the background infrastructure through a mobile phone. Therefore, the latter scenario requires a slightly different approach described in subsection IV-C.

A. Using Bluetooth-enabled Active Tags to Interact with Mobile Phones

Mobile phones are likely to play an important role in future pervasive computing applications because they have become economically very successful and almost ubiquitous. In Europe, they are the only mobile technology that is widely accepted. As an increasing number of mobile phones with integrated Bluetooth support appear on the market, BTnodes and therefore the smart objects they are attached to can make direct use of mobile phone features.

In the scenarios (cf. section III), mobile phones are used in the following ways:

- BTnodes communicate with background infrastructure services by using phones as mobile access points. Furthermore, they store interaction stubs on the devices that are used in further interactions. These kinds of communications take place invisibly for the user.
- Mobile phones are used by smart objects to notify people via alarms, SMS messages and custom calls or by sending OBEX objects, like calendar entries and business cards, to the phone.

- Interaction stubs stored on mobile phones can be used by users to explicitly initiate interactions with smart objects or their virtual counterparts from everywhere.



Fig. 9. A business card and a calendar entry transferred to a mobile phone from a BTnode.

Mobile phones offer some interesting features for implementing the actual user interface for the interaction with smart objects. Those features range from SMS messages over alarms to OBEX objects and Java applications. Over the Bluetooth serial port profile (SPP) and dialup networking profile (DUN), AT commands can be sent to mobile phones. A standard set of AT commands is supported by virtually all GSM phones, although some manufacturers support a wide range of additional commands. AT commands are used to send and receive SMS messages, to store SMS templates, to handle alarms, and for building up a connection to the background infrastructure.

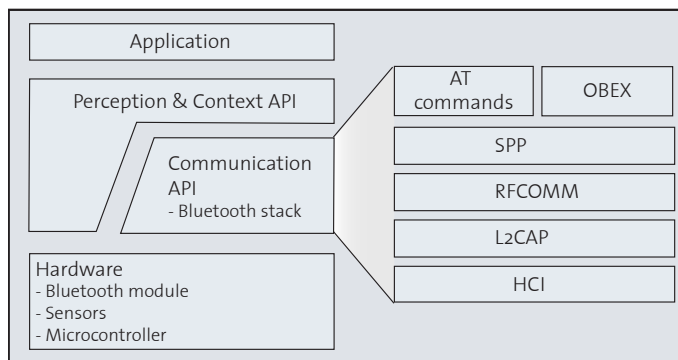


Fig. 10. Software structure of the BTnodes and of the Bluetooth stack. HCI (Host Controller Interface), L2CAP (Logical Link Control and Adaptation Protocol), RFCOMM, SPP (Serial Port Profile), and Bluetooth OBEX (Bluetooth Object Exchange Profile) are specified by the Bluetooth standard.

Another possibility for smart objects to interact with mobile phone users, is to send OBEX objects such as business cards and calendar entries to mobile phones. Transmission of business cards usually does not require authentication, and is a suitable way to send short notifications to users. However, due to the structure imposed by the vcard standard, it is not a very intuitive way for interaction. Calendar entries can be used by a wide range of smart objects to remind people. Figure 9 shows

examples of a business card and an calendar entry transmitted by a BTnode. We have written a Bluetooth stack for the BTnodes (i.e. on the Atmel ATmega128 microcontroller) that supports transmission of AT commands and OBEX objects to mobile phones. The structure of the stack is depicted in figure 10.

Even if a user's mobile device does not support Bluetooth, the BTnodes can still be used to communicate with them via the cellular phone network. However, this requires a GSM gateway with a Bluetooth access point in range of a smart object. This way, SMS messages can still be sent to the user's phone as long as its number is known to the smart objects.

B. Embedding Smart Objects into the Everyday Communication Infrastructure

The following requirements must be fulfilled to implement the interaction patterns described in the remote interaction and smart product monitoring scenarios:

- 1) A smart object, i.e. a BTnode attached to it, must be able to initiate and receive calls over the cellular phone network at any time, independent from the user's current location.
- 2) Users as well as devices and their service parameters must be determinable according to the information stored on RFID labels attached to them. Service parameters describe how devices can be accessed.
- 3) An active tag must be able to derive its own context and the context of nearby people that want to interact with a smart object.
- 4) A BTnode must possess the capability to transmit interaction stubs to mobile phones.

In order to ensure requirement (1), a smart object is assigned a fixed telephone number. This is done by a GSM gateway that is accessible from a network existing between smart devices. Beutel et al. [3] describes how data packets can be routed in an ad hoc network of BTnodes over multiple hops. In our implementation, the GSM gateway consists of a mobile phone and a Bluetooth module attached to a laptop PC with Internet connection, which also serves as a stationary Bluetooth access point for the smart objects (cf. figure 11). The telephone number of smart objects that are connected to the access point is the phone number of the mobile phone serving as GSM gateway. The access point is responsible for relaying incoming calls and messages to the correct smart object. Currently, this is done based on the content of incoming messages. When a smart object needs to call a user's phone this is also done through the mobile phone serving as GSM gateway. In order to handle calls, i.e. in order to send and receive SMS messages, the software of the access point sends AT commands to the mobile phone.

Data packets from smart devices are sent over a Bluetooth network to the access point. A Bluetooth bridge that we have written for packets originating from BTnodes writes incoming data packets into a tuple space. The TSpaces software package from IBM [28] was used as tuple space implementation and the BlueZ stack [6] as Bluetooth stack for the access point. The tuple space, which can be regarded as a service of the background infrastructure, serves as storage medium for the BTnodes.

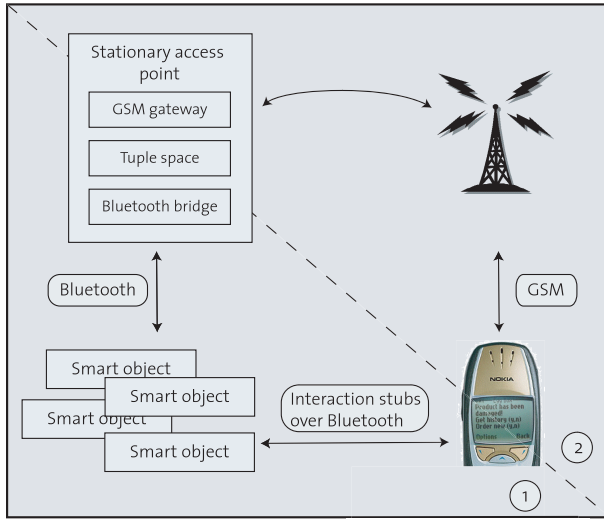


Fig. 11. Overview of the main architectural components in the remote interaction and smart product monitoring scenario: when the user is in range of a smart object, interaction stubs are transferred to his/her mobile phone (1), when far away, communication takes place over the cellular phone network (2).

Also, more elaborate services for the smart objects are implemented by using the tuple space. For example, a BTnode is allowed to register a callback function within the tuple space that is executed when a certain kind of tuple is derived. For instance, when there exists a tuple containing the phone number of a worker in charge of certain smart products together with a tuple indicating that one of this products has been damaged, a corresponding result tuple is derived and its associated callback is executed. This function sends an SMS message as notification to the worker's mobile phone.

Requirement (2) is met by facilitating wireless RFID scanners that transmit RFID data via Bluetooth to smart objects. We built wireless RFID scanners based on BTnodes and Bluetooth-enabled PDAs. When a tag is scanned by an RFID reader attached to a BTnode or a PDA, the corresponding information is transmitted to the access point and stored in the tuple space.

The RFID labels store information of how to access devices or data to identify people. For example, Bluetooth enabled mobile phones are tagged with an RFID tag containing their phone number and Bluetooth device address (BD_ADDR), RFID tags attached to BTnodes store their BD_ADDR, mobile phones not supporting Bluetooth have an RFID tag attached containing their GSM phone number. The main advantage of this approach is that we allow smart objects to interact with devices that do not support their radio technology or communication standard. This way, BTnodes can interact with users whose mobile phones do not support Bluetooth. RFID labels become the lowest level for ensuring that interaction can take place in smart environments. Because of their low cost and wide availability they are the ideal technology for this purpose.

Regarding requirement (3), BTnodes have the ability to perceive their environment through sensors, such as temperature and acceleration sensors or RFID scanners. BTnodes derive context information on the basis of locally stored information (e.g. the last known state of an object), their own sensory input (e.g. acceleration measurements), and data that they obtain

from the access point and other BTnodes in their proximity. In the smart product monitoring scenario (cf. section III), the context "state of a smart product" is derived on the basis of the BTnode's own sensory input and locally stored information: What is the last known state of the product? What are the lowest and highest temperature and acceleration values the product can withstand? What is the current temperature and acceleration?

However, context that can be computed by a single BTnode is not sufficient to implement the described interaction patterns. In order to find out who shares a certain symbolic location with a smart object, for example, collaboration with other smart objects and the access point becomes necessary. Users and their mobile devices are identified by RFID labels. When they enter a symbolic location, such as an office, their ID or their service parameters are sensed by a BTnode with attached RFID scanner and sent to other smart objects and the access point. Only by considering this information, a smart object can decide who is in the same room. Radio connectivity of the communication module is not suited to derive this kind of context information.

In the described scenarios, interaction stubs enable users to interact with smart objects remotely and must therefore be transmitted to user's mobile phones (requirement (4)). Regarding the smart product monitoring scenario, the interaction stubs consist of the phone number of a smart product and the commands that can be processed by it. These information are transferred to the user's mobile phone together with an SMS message. Because the smart product initiates the interaction and sends the first message, its phone number is implicitly given. The user just needs to reply to the message. The commands that are supported by the smart object are also embedded in this message.

Concerning the remote interaction scenario, interaction stubs are transferred to mobile phones unnoticed by human users. Here, the last requirement can only be fulfilled when users have Bluetooth-enabled mobile devices, because the transmission of interaction stubs (phone book entries and SMS templates) requires a Bluetooth connection to the phone over which AT commands can be transferred. The interaction stubs in this scenario are the phone book entries for smart objects and the SMS templates. By using the phone book entries, smart objects can be addressed remotely. The SMS templates contain the commands a smart object understands.

C. Mobile Infrastructure Access Points and Virtual Counterparts

Applications that make use of passive RFID technology and active tags often require access to background infrastructure services because the information stored on RFID tags, such as electronic product codes, cannot be semantically interpreted by a peer-to-peer network of smart objects, but requires access to large databases. Examples of possible requests to background infrastructure services are: What is the name of the medicine with product code p ? When and how often must a patient take the medicine with product code p ? When in a medicine cabinet are different kinds of medicine with product codes p_1, \dots, p_n , may a patient take a medicine with the product code p ? Although it is possible to access background infrastructure ser-

vices through a stationary PC with Internet connection, the assumption that there is always such a PC running in range of smart objects in today's environments is very strong.

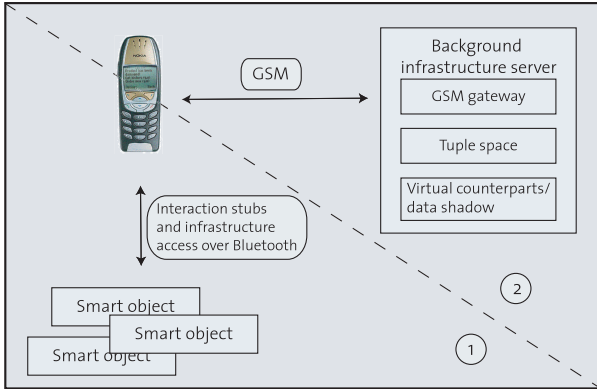


Fig. 12. Overview of the main architectural components for the smart medicine cabinet scenario. Smart objects use mobile phones to store interaction stubs and to communicate with background infrastructure services when a user is in range of the object (1), when away from the object, requests from the user are handled by the virtual counterpart (2).

Therefore, in the smart medicine cabinet scenario, background infrastructure services are accessed by a smart object through the user's mobile phone via the cellular phone network. The advantage of this approach is that there is no need for extra equipment. On the other hand, a smart object can only access the infrastructure when there is a user in range of the object. If a person is not in range of the object he/she cannot communicate with it directly, which makes it necessary to have a representation of this object present in the background infrastructure, which processes requests to the smart object, as long as it cannot be addressed directly. This representation is called the *virtual counterpart* [14] or the *data shadow* [7] of the object. The state of a smart object, i.e. of a BTnode, and its virtual counterpart are synchronized each time the object accesses the background infrastructure through a mobile phone. Figure 12 shows an overview of the main components for the smart medicine cabinet scenario.

In our implementation, access to the background infrastructure is implemented by sending SMS messages from a BTnode to a background infrastructure server with GSM gateway. The GSM number of this gateway is predefined in the BTnode's software, and the SMS message with the embedded commands for the background infrastructure server are sent by transmitting AT commands over a Bluetooth connection from a BTnode to the user's mobile phone.

When a BTnode receives the response to its query, it can decide about the semantics of a product code, e.g. when and how often a patient has to take the medicine and stores corresponding alarms to the user's mobile phone. This is again done by transferring AT commands over Bluetooth to the phone. Alternatively, corresponding calendar entries can be transmitted to the phone. Also, interaction stubs, i.e. a phone book entry and an SMS template containing human readable commands for the virtual counterpart are sent to the user's phone by using AT commands. The phone book entry, however, does not contain the number of the smart object. The smart object is

assigned a phone number only temporarily when a user with his/her mobile phone is in range of the object. Then, the number of the user's phone is also the number of the smart object, and it is used by the background infrastructure server to contact the BTnode. But when no mobile phone is in Bluetooth range of the smart object it has no phone number. Therefore, the phone number of the GSM gateway of the background infrastructure server is stored as phone book entry in the user's mobile phone. When far away from the medicine cabinet, the requests are then sent via SMS messages to the background infrastructure server, which processes embedded commands.

V. RELATED WORK

Schmidt [20] coined the term implicit human computer interaction and proposed an XML-based language for describing implicit interaction in ubiquitous computing environments. First experiences with computer-augmented everyday artifacts are described by Beigl et al. [2]. In the MediaCup project [12] [17] active tags were attached to coffee cups, and information derived by these tags was used to build context-aware applications. Several other projects developed context-aware systems, e.g. the GUIDE [9] or Active Badge [26] projects. Holmquist et al. [13] uses accelerometer measurements to associate smart artefacts by shaking them together. Ringwald [18] investigated explicit association in smart environments with laser pointers.

The concept of a medicine cabinet augmented by information technology has been demonstrated previously by Wan [25]. The focus of the implementation by Wan has been to create a "situated healthcare portal" in the bathroom by integrating a personal computer, an LCD screen and a broadband Internet connection into the medicine cabinet. The medicine cabinet presented as part of our work was designed with the goal to leave the medicine cabinet practically unmodified from a user perspective.

VI. CONCLUSION

This paper investigated interaction patterns in pervasive computing settings where Bluetooth-enabled active tags and RFID labels are attached to everyday objects and products. In these environments hybrid approaches for the association of interaction partners become important. By using the concept of invisible preselection, interaction stubs that also enable remote interaction with smart objects are downloaded to mobile devices subject to the current context of users. Well known mobile phone features, e.g. SMS messages, phone book and calendar entries, that are familiar to a vast majority of users are well suited to interact with smart devices independent from the user's current location. Two architectures were described that enable remote interaction. The first assigns fixed phone numbers to smart objects and allows users to always communicate with their items directly. The second architecture assigns phone numbers only temporarily to smart objects. Here, mobile phones are used as mobile access points for smart objects that allow communication with background infrastructure services and synchronization with virtual counterparts. When there is no direct connection to a smart object, queries are processed

by virtual counterparts. Active and passive tags are complementary to each other. By attaching RFID scanners to active tags and their integration in everyday objects we were able to combine both technologies. This approach enables novel applications in which technology truly disappears.

REFERENCES

- [1] The MIT Auto-ID Center. <http://www.autoidcenter.org/>.
- [2] M. Beigl, H.-W. Gellersen, A. Schmidt: Mediacups: Experience with Design and Use of Computer-Augmented Everyday Artefacts. *Computer Networks*, Special Issue on Pervasive Computing, Elsevier, Vol. 35, No. 4, March 2001.
- [3] J. Beutel, M. Dyer, O. Kasten, M. Ringwald, F. Siegemund, L. Thiele: Bluetooth Smart Nodes for Mobile Ad-hoc Networks. Submitted for publication, October 2002.
- [4] J. Beutel, O. Kasten: A Minimal Bluetooth-Based Computing and Communication Platform. Technical Note, May 2001.
- [5] Bluetooth Special Interest Group: Specification of the Bluetooth System, Core and Profiles. <http://www.bluetooth.com>, Version 1.1, February 2001.
- [6] BlueZ - Official Linux Bluetooth Protocol Stack. <http://bluez.sourceforge.net/>.
- [7] The cooltown project: <http://www.cooltown.com>.
- [8] A. Davie: Intelligent tagging for transport and logistics: the ParcelCall approach. *Electronics & Communication Engineering Journal*, Volume 14, Issue 3, June 2002.
- [9] N. Davies, K. Cheverst, K. Mitchell, A. Friday: Caches in the Air: Disseminating Information in the Guide System. *Proc. of the 2nd IEEE Workshop on Mobile Computing Systems and Applications (WM-CSA99)*, New Orleans, Louisiana, USA, February 1999.
- [10] E. Fleisch, F. Mattern, H. Österle: Betriebliche Anwendungen mobiler Technologien: Ubiquitous Commerce. *Computerwoche*, Collaborative Commerce, February 2002.
- [11] Forrester Research: <http://www.forrester.com>.
- [12] H.-W. Gellersen, M. Beigl, H. Krull: The MediaCup: Awareness Technology embedded in an Everyday Object. 1st International Symposium on Handheld and Ubiquitous Computing (HUC99), Karlsruhe, Germany, 1999.
- [13] L. E. Holmquist, F. Mattern, B. Schiele, P. Alahuhta, M. Beigl, H.-W. Gellersen: Smart-Its Friends: A Technique for Users to Easily Establish Connections between Smart Artefacts. *Proc. Ubicomp 2001*, Springer-Verlag LNCS 2201, pp. 116-122, 2001.
- [14] M. Langheinrich, F. Mattern, K. Römer, H. Vogt: First Steps Towards an Event-Based Infrastructure for Smart Things. *Ubiquitous Computing Workshop (PACT 2000)*, Philadelphia, PA, October 2000.
- [15] C. H. Marcussen: Comparing SMS and WAP in Europe with i-mode in Japan. *Next Generation WAP Services and i-mode*, London, UK, 2002.
- [16] F. Mattern: The Vision and Technical Foundations of Ubiquitous Computing. In: *Upgrade European Online Magazine*, pp. 5-8, October 2001.
- [17] The MediaCup project: <http://mediacup.teco.edu>.
- [18] M. Ringwald: Spontaneous Interaction with Everyday Devices Using a PDA. *Proceedings Workshop on Supporting Spontaneous Interaction in Ubiquitous Computing Settings, Ubicomp 2002*, Gothenburg, Sweden, September 2002.
- [19] B. Schilit, N. Adams, R. Want: Context-aware computing applications. *Proceedings of IEEE Workshop on Mobile Computing Systems and Applications*, Santa Cruz, California, December 1994.
- [20] A. Schmidt: Implicit Human Computer Interaction Through Context. *Personal Technologies*, Volume 4(2&3), June 2000.
- [21] F. Siegemund: Kontextbasierte Bluetooth-Scatternetz-Formierung in ubiquitären Systemen. *Proceedings 1st German Workshop on Mobile Ad-Hoc Networks*, Ulm, Germany, March 2002.
- [22] F. Siegemund, M. Rohs: Rendezvous Layer Protocols for Bluetooth-Enabled Smart Devices. *Proceedings 1st International Conference on Architecture of Computing Systems - Trends in Network and Pervasive Computing, ARCS 2002*, Karlsruhe, Germany, April 2002.
- [23] The Smart-Its Project. <http://www.smart-its.org/>.
- [24] Texas Instruments: Texas Instruments RFID Technology Enables M-Commerce Cashless Payments for Nokia Phone Users. <http://www.ti.com/snc/docs/news/rel32.htm>
- [25] D. Wan: Magic Medicine Cabinet: A Situated Portal for Consumer Healthcare. *Proceedings of the International Symposium on Handheld and Ubiquitous Computing*, Karlsruhe, Germany, September 1999.
- [26] R. Want, A. Hopper, V. Falcao, J. Gibbons: The Active Badge Location System, *ACM Transactions on Information Systems* 10(1), January 1992, pp. 91-102.
- [27] M. Weiser: The Computer for the Twenty-First Century. In: *Scientific American*, pp. 94-100, September 1991.
- [28] P. Wyckoff: TSpaces. *IBM Systems Journal*, Volume 37, Number 3, 1998.