The Digital Sommelier: Interacting with Intelligent Products

Michael Schmitz[1,2], Jörg Baus[1,2], and Robert Dörr[1]

[1] DFG - Collaborative Research Centre
Resource-adaptive cognitive processes
Saarland University
PO-Box 15 11 50, Germany
{schmitz,baus,rdoerr}@cs.uni-sb.de
[2] DFKI GmbH
Stuhlsatzenhausweg 3 (Building D3 2)
66123 Saarbrücken, Germany
{michael.schmitz,joerg.baus}@dfki.de

Abstract. We present the *Digital Sommelier*, an interactive wine shopping assistant that provides an intuitive multi-modal interface to general product information as well as to particular attributes of a certain product, such as its current temperature. Wine bottles sense their state via attached wireless sensors and detect user interaction over RFID and acceleration sensors. Visitors can inquire information either through physical interaction with products or a natural language interface. We describe a framework and toolkit for efficient prototyping of sensor based applications as the foundation for the integration of different sensor networks utilized by the sommelier. We further introduce our concept of talking products, an anthropomorphic interaction pattern that allows customers to directly talk to products with personalities.

1 Introduction

Emerging sensor technologies that go beyond RFID recognition are becoming cheaper and smaller, paving their way into objects of everyday life. The potential ubiquity of wireless sensors will particularly affect future generations of products as elements of production, processing and delivery chains, enabling them to sense and record their state and communicate with their environment. The prospective immediate benefits for manufacture and retail are an essential factor that is increasing the chances for research work in this field to get adopted by industry - just as RFID technology has already made the step into the commercial sector, as it provides economic values by facilitating process automatization such as product tracking or inventory stocktaking. The end consumer does not directly benefit from this technology, as long as the introduction of RFID technology does not transparently result into price reductions. Sensoric and communication technology that continously sense and observe the state of a product open up possibilities for new services that also benefit end consumers: One simple example is quality assurance of sensitive food such as fresh fish, by monitoring

C. Floerkemeier et al. (Eds.): IOT 2008, LNCS 4952, pp. 247–262, 2008.
© Springer-Verlag Berlin Heidelberg 2008

the continous maintenance of the cold chain. By providing information about the temperature history to end customers, their trust into the product could be increased. Temperature is only one product attribute that could provide useful information to the customer: We envision that products will be able to record a complete digital diary with all relevant information, particularly regarding their quality and operation. Being able to communicate with their environment it is further important to allow humans to access such a product memory.

We developed a demonstration application, the *Digital Sommelier* (DigiSom), that provides a simple interface for customers to retrieve general features of the product (wine and champaign bottles) as well as specific state attributes: the current temperature and whether the bottle has been recently shaken. Our aim was to support novice users in their shopping activities with an intuitive interface. The system recognizes user interactions based on RFID and sensor technology, and provides a natural language interface combined with visual output of information. We also integrated our concept of anthropomorphic products into this scenario: Each type of wine is associated with a distinct personality, which is reflected by prosodic attributes of the voice and by different phrasing styles. The implementation of the sensor integration is based on a framework that we have designed for rapid development of sensor based applications. In this work we present first results of our group in this field: The framework for sensor based applications, the anthropomorphic interaction pattern and the DigiSom, an application integrating both concepts in a demonstration system that has been exhibited on various occasions.

The remainder of this paper is organized as follows. First, we describe the architecture of our framework for sensor based development in chapter 2. Chapter 3 covers the conceptual and technical background of the talking products paradigm. An overview of the DigiSom installation is given in chapter 4 and chapter 5 concludes this paper with a brief discussion.

2 A Framework for Sensor Based Prototyping

Embedding sensing, computing and communication capabilities into physical artifacts is the essential building block for the creation of intelligent objects. Some simple examples can already be found in commercial products: Apples iPhone for instance uses an acceleration sensor for flipping images, a proximity sensor for switching off the display when speaking and a light sensor for adjusting brightness. The latter technique is also used in some desktop monitors. Such standalone applications use their collected data only for one distinct and straightforward purpose and do not communicate it to other devices. In a next step we envision networks of things, where all sorts of information about objects are accessible and exchangeable between artifacts and/or the environment.

Research in the field of wireless sensor networks has brought up commercial off-the-shelf systems, which provide sensor nodes with integrated sensing, computing and communication capabilities. Some of them can be configured only by means of basic parameters, others have a programmable microcontroller and can

be flashed with custom firmware. Such devices use their computing resources for more sophisticated tasks like preprocessing sensor readings or routing of messages through the network - even applications running completely inside the network are possible.

Advances in miniaturization allow such devices to be included in a wide range of objects. In our intelligent environment [1], we use wireless sensor nodes to monitor general environmental parameters like temperature or light intensity, as well as to observe the state of objects (i.e. products).

2.1 Sensor Platforms

No single sensor platform can meet all the requirements of different applications. Therefore we work with various wireless sensor platforms, namely the μPart, the MICAz and Scatterweb, while using the latter system only for environmental observations, such as measuring the temperature in a room or detecting the presence of people. The μPart platform has been initially developed at Karlsruhe University [2]. It is produced and sold by the spin off Particle Computer GmbH. μPart nodes provide sensors for light, movement and temperature. They can be configured to set the sampling rate and other basic parameters through light sensors, but firmware modifications are not possible this way. The MICAz platform is a Zigbee compliant implementation of the Mica Mote platform [3], developed at Berkeley University in cooperation with Intel. MICAz nodes are sold by Crossbow Inc. and are shipped without integrated sensors, but they can be stacked with various sensor boards. We use them mainly in connection with the MTS300 board, which provides sensors for light, temperature and noise as well as a dual-axis accelerometer and a dual-axis magnetometer. The ATmega128L microcontroller on the nodes provides computing resources to run custom firmware from the integrated 128 KBytes of flash memory.

2.2 Framework Requirements

So far applications have to access sensor data directly through hardware dependent libraries. In many cases hardware details influence the entire software design, which complicates software maintenance especially when hardware is exchanged or updated. This particularly poses a burden in research environments, where different device platforms are interchanged and used for different applications, reusability of software modules is very restricted.

To enhance rapid development of sensor based applications, a solution had to be found, which provides a structure for data processing through reusable software components. From previous experience with sensor applications we have extracted the following requirements for such a framework:

1. Sensor data should be accessible through a high level API instead of raw byte access.
2. Data processing should be as simple and efficient as possible to cope with high data rates from a large number of sensors simultanously.

3. All sensor specific hardware and software details should be hidden.
4. A large number of data sources should be supported, including not only wireless sensor networks but also other types of input (e.g. reading from files, databases or other network sources).
5. There should be prebuilt components for common application tasks as well as for development related issues (e.g. data inspection, debugging).
6. The implementation should be open for extension, allowing further development and adaption of new sensor hardware.

2.3 Processing Sensor Data Streams

The design of the framework is based on the concept of stream processing [4] (see [5] for an overview). It uses a data driven approach where information is contained in possibly infinite streams of data entities. Each entity can be seen as an event and in our terms, each raw sensor reading is such an event. In our framework, events are produced, consumed and processed by three kinds of modules: *Input modules* encapsulate data sources such as sensor networks or files, and provide a steady flow of information into the framework. *Output modules* are for example used for displaying data, connecting to other applications or triggering actions based on events. Inbetween input and output optional *filtering modules* process the data streams to extract and prepare relevant and meaningful information from incoming sensor data. The data flow is implicitly uni-directional - if for any reason a bi-directional flow is required, it must be explicitly constructed.

The sensor based component of a new application is built by choosing one or more modules of a set of modules provided by the framework and connecting their in- and outputs appropriately to construct a desired function. Although new custom modules can be easily implemented and integrated, the set of prebuilt modules covers many common data processing tasks, such that a wide range of simple applications can be developed quickly with the given components.

As sensor nodes define their own message format, the encapsulating input modules are customized to suit the underlying sensor platform, and have to be extended, whenever new platforms have to be integrated. Such input modules provide flexibility and connectivity towards the framework, a back channel to inject messages into a sensor network is not implemented yet, since the main focus of the framework is to simplify data extraction and processing from sensors.

There are already several tools available to support event processing for intelligent environments. We are using the EventHeap [6] to share data between applications, but sending each sensor reading directly to the EventHeap is certainly not a good idea, since a single reading does not contain much information. The raw sensor readings have to be preprocessed and analyzed in order to generate higher level events that are meaningful enough to be sent to the actual application. The Context Toolkit [7] provides a well structured framework for working with context information. Its Generator-Widgets encapsulate context information gained from sensor readings, but a concept to extract high level information from raw sensor data is not provided.

2.4 Implementation

We decided to implement the framework in Java mainly because of two reasons: Most sensor networks already come with some Java library or classes giving a good starting point. Furthermore, Java is platform independent, which is particularly important when working in a heterogenous environment.

Data entities which are transmitted through the framework are represented by the *Message* class (fig. 1). Each message contains the source id, a timestamp, the data value and a type field. All fields are Strings, except of the timestamp which is of type long.

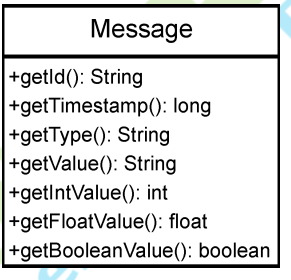


Fig. 1. The Message class (part)

As already stated, there are three types of modules, namely *Input*-, *Output*- and *Filter*-Modules. They are linked together using a design pattern from Object Oriented Programming [8] called the Listener Pattern. It allows coupling of software components and uni-directional transmission of data from a Subject (Input) to one or more Listeners (Output). Filter modules implement both, `StreamIn` and `StreamOut` interfaces and are further divided in buffered und unbuffered filters. The original pattern has been extended by a tagged version to allow Output streams being grouped together. Figure 2 shows the main interfaces.

This approach allows to seperate the application architecture from the specific implementation of modules. The architecture can be described in a model-driven fashion using block diagrams, which contain the modules as well as their connections. Figure 3 shows a diagram for a simple application based on prebuilt modules. It takes sensor data from μPart nodes, filters out everything but temperature readings, and writes them to a file only when sensor value changes. The architecture can be easily implemented as shown in listing 1.1. Each box corresponds to a constructor call and each arrow corresponds to a call of the `addOutputStream()` method. More complex applications may incorporate multiple sensor networks, more sophisticated filtering and different output modalities.

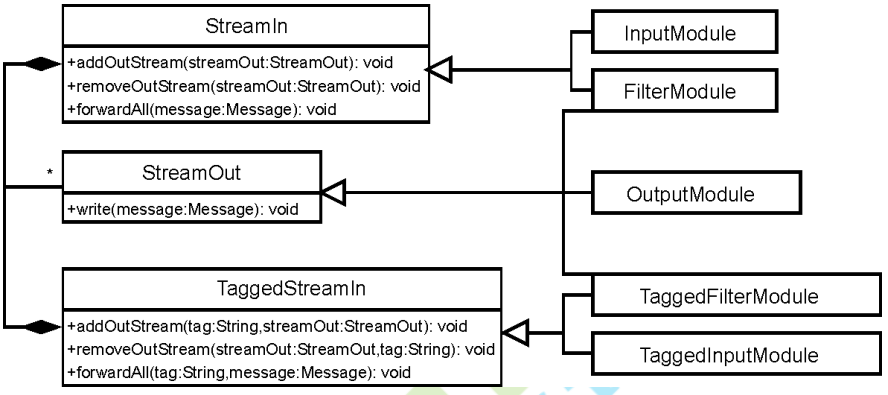


Fig. 2. Extended Listener Pattern

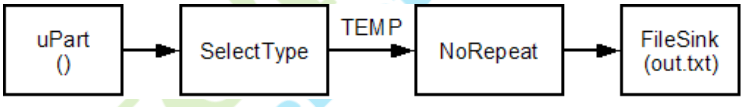


Fig. 3. Block diagramm for an application that writes sensor readings to a file

```
// construct modules
UPart input = new UPart(); // use default port
SelectType selectType = new SelectType();
NoRepeat noRepeat = new NoRepeat();
FileSink output = new FileSink(new File("out.txt"));

// link modules
input.addOutputStream(selectType);
selectType.addOutputStream("TEMP", noRepeat);
noRepeat.addOutputStream(output);

// start receiving data
input.start();
```

Listing 1.1. Implementation of the application described by figure 3

So far, there are about 50 prebuilt modules. Input can be obtained from different sensor networks, from files or from network or serial connections. Amongst others, output can be directly fused into files, databases or higher level systems like the EventHeap. For data inspection and debugging there are modules that display data either graphical of textual. Filter modules implement various computing functions like sum, variance or standard deviation of a sequence of

values. There are filters that drop repeating values, split message streams by origin or type, translate certain field values of the messages and many more. The translation for example is often used to map sensor IDs to products and can be configured through an external XML file, eliminating the need for touching the binaries when changing product mappings. Modules are implemented by extending abstract base classes, which already contain functionality for establishing links between modules, so there is minimal effort needed (see listing 1.2 for an example).

```
public class EvenOddFilter extends Filter {
    protected Message filter(Message msg) {
        try {
            if(msg.getIntValue() % 2 == 0) {
                msg.setValue("Even");
            } else {
                msg.setValue("Odd");
            }
            return msg;
        } catch (NumberFormatException e) {
            return null;
        }
    }
}
```

Listing 1.2. Implementation of an example filter module

2.5 Special Purpose Modules

Not all algorithmic problems can be broken down into simple solutions. In some cases it is more reasonable to implement a complex function as a self contained module, which provides the necessary interfaces for connecting it to the framework. Robust movement analysis based on acceleration sensors is such a task, requiring sophisticated methods for information extraction. The complexity of the given data in this case requires a tailored approach, for which we generated a specialized module for our framework.

Acceleration data is sampled with a rate of 100 samples per second. In a first step the readings are preprocessed by computing several statistical values over a window of 30 samples corresponding to about 300ms. The statistical values are mean, variance, standard deviation as well as the number of local minima and maxima for each acceleration axis, forming a block of 10 characteristic values, which describe the state of acceleration for a specific time interval. These blocks are computed by our own firmware installed on the sensor boards, and are sent to the receiving PC that runs the framework and then processed by WEKA [9], first in a training phase and after that in a classification phase. In the training phase, several classes of movement states have to be specified, and incoming readings

are assigned to the according movement class. Based on this training data the system can classify incoming data packages and send results further into the framework. Using our statistical approach of movement state classification, we are able to differentiate the movements regarding their intensity (no movement, slow movement, fast movement), their direction or anything else that can be extracted from the set of statistical values. The classification is done by the WEKA implementation of a Ripple-Down Rules (RDR) classifier [10], but other classifiers like C4.5 decision trees [11] have also proven to be useful. For a user of our framework, there is no need to write a single line of code during this process. A self written tool allows the training phase to be completed in a few minutes just by moving the sensor, clicking buttons and entering names for different movement classes.

2.6 Integrating Sensors into the Digital Sommelier

There are two different tasks for our sensor equipment in the DigiSom application: Collecting and storing sensor readings that define the state of wines (temperature and vibration) using the μPart nodes, and acceleration sensors on the MICAz nodes to detect, whether a user looks at the front or at the rear side of a product box. Figure 4 shows the design of the sensor processing application used for the DigiSom.

Temperature processing was a rather straightforward task, there were only two minor issues which have to be dealt with. First, the μParts do not allow to set the sampling rate of temperature and vibration sensor independent of each other. To provide adequate reaction times in detecting vibration, we had to choose a sampling interval of about one second. This results in a steady stream of equal temperature values, which can be limited by the *NoRepeat* filter module.

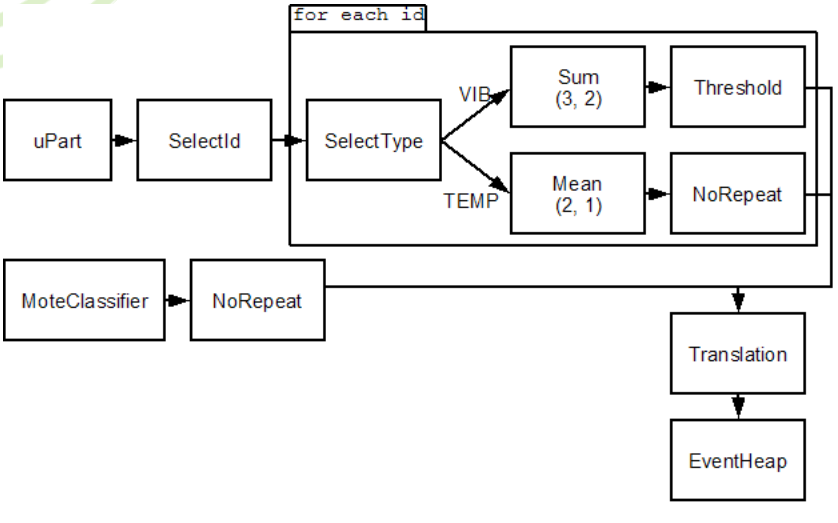


Fig. 4. Block diagramm of sensor data processing in the DigiSom

The second problem is that at the transition between two degree values, there is some interval of jittering, where sensor readings continously switch between these values. To prevent these messages from slipping through the NoRepeat filter, we compute the mean of two successive readings. This is the simplest form of a moving average represented by the module *Mean* with a buffer size of two and with an overlap between buffers of one message (see Figure 4). For vibration detection the μPart nodes use an integrated ball switch. This is great for implementing a wakeup on move function, but gives only poor results for deciding, if a bottle has been shaken or has just been moved. Vibration readings are summed up over a buffer of three messages with an overlap of two. The result is then compared to a threshold, which represents the sensitivity. To detect the orientation of the product boxes, we integrated MICAz nodes. Their two-axis accelerometer measures the acceleration caused by gravitation. A user looking at a product never holds it perfectly upright, but always in a certain angle such that she must not hold the box at eye level for a proper view. Taking this observation into account, we have trained our nodes to classify front, back and upright orientation. The classifier forwards its results into the framework, where repeating messages are dropped. After preprocessing is done, messages are translated to map sensor node IDs to poducts and published as EventHeap tuples to make them available to other parts of the DigiSom.

3 Dialogue Shell of Talking Products

One important design goal of our interactive shopping assistance is to support arbitrary users, particularly computer novices, who are not able or willing to learn the use of such a system. We therefore have to find a solution that provides a natural interaction, requiring minimal effort of a user to understand and utilize the assistance system. Nijholt et al. [12] suggest that a limited animistic design metaphor seems to be appropriate for human-environment interaction with thousands of networked smart objects. People often tend to treat objects similar to humans, according to findings of Reeves and Nass [13], which allows users to explain the behavior of a system if they lack a good functional conceptual model. In consequence, we decided to employ a natural language system, which enables the user to talk to each product.

[14] conducted a usability study of a multi-modal shopping assistant. The implemented system allows users for instance to request product information in a combination of speech and selecting gestures (i.e. taking a product out of the shelf). Findings of this study showed among others that users generally preferred direct over indirect interaction, i.e. by asking "What is your price?" instead of "What is the price of this camera?", which encouraged us to pursue this approach.

Previous studies have shown that interacting with embodied conversational agents that have consistent personalities is not only more fun but also lets users perceive such agents as more useful than agents without (consistent) personalities [15,16]. It is further shown that the speech of a consistent personality enables the listener to memorize spoken contents easier and moreover reduces the overall

cognitive load [17,15]. Thus we emphasized the anthropomorphic aspect of this interaction pattern by assigning personalites to products, which are reflected by the spoken responses of a product.

Product manufacturers benefit as well, since the personalization of the product provides a new channel to communicate a brand image or distinct attributes of a certain product. A study within the context of marketing research showed that if in radio advertisements a voice fits the product, it helps the listener to remember the brand, the product and claims for that product [18].

3.1 Modelling Personality in Voices

In a first step we created voices that reflect certain personalities according to Aaker's brand personality model [19] only by adjusting prosodic parameters. We chose this model over the (rather similar) five factor model [20] commonly used in psychology, since we are applying the concept of talking objects in the shopping domain. However, both models are rather similar and to a certain extent exchangable.

We changed the four prosodic parameters pitch range (in semitones), pitch level (in Hz), tempo (as a durational factor in ms) and intensity (soft, modal or loud as in [21]) according to our literature review [22]. For example, a competent voice has a higher pitch range (8 semitones), a lower pitch level (-30%), a 30% higher tempo and a loud intesity compared to the baseline voice. In [22] we also evaluated whether it is possible to model different personalities with the same voice by adjusting these prosodic parameters, such that listeners will recognize the intended personality dimension. The study has shown that there are clear preferences for our prosody modeled speech synthesis for certain brand personality dimensions. But not all personality dimensions were perfectly perceived as intended, such that we have to amplify the effect.

Personality is certainly not only expressed in qualitative attributes of a voice, other properties of a speech dialogue are also essential, like the used vocabulary or the general discussion behaviour. For this reason we created a dialogue shell that incorporates these aspects.

3.2 Expressing Personality in Dialogues

The widely adopted personality model by Costa and McRae [20] constitutes five dimensions of human personality: Extraversion, Agreeableness, Conscientousness, Neuroticism and Openness on a scale from 0 to 100. Obviously, differentiating 100 levels in a dimension is far too much for our goals, therefore we simplified this model by discriminating 3 levels in each dimension:

- low: value between 1 and 44 (31% of population)
- average: values between 45 and 55 (38% of population)
- high: values between 56 and 100 (31% of population)

Related work, e.g. by Andre et al. [23] limited their personality modelling to only two of the five dimension - namely extraversion and agreeableness - since

these are the most important factors in interpersonal communication. Nevertheless, we discovered considerable influences of openness and conscientiousness to speech, therefore we incorporated these two dimensions as well. The effect of the dimension neuroticism is mainly to describe the level of susceptibility to strong emotions, both positive and negative ones [24]. It is further shown that the level of neuroticism is very hard to determine in an observed person [25], thus we decided that four dimensions will suffice for our work.

We conducted an exhaustive literature review on how speech reveals different personality characteristics. Among numerous other ressources, two recent research papers provided essential contributions to our work: Pennebaker and Kings analysis in *Journals of Personality and Social Psychology* [26] and Nowson's *The Language of Weblogs: A Study of Genre and Individual Differences* [27]. In both studies a large number of text blocks were examined with an application called *Linguistic Inquiry and Word Count*[1] (LIWC), which analyzes text passages word by word, comparing them with an internal dictionary. This dictionary is divided in 70 hierarchical dimensions, including grammatical categories (e.g. noun, verb) or affective and emotional processes. Pennebaker determined in a study the 15 most reliable dimensions and searched for them in diary entries of test persons with LIWC. With these results together with the given personality profiles of the probands (according to the five factor modell), he identified correlations between the two. Nowson performed a similar study and searched through weblogs for the same LIWC factors.

Based on these results, we provided a set of recommendations how responses of an talking object with a given personality should be phrased. For instance, for a high level of extraversion these recommendations are given:

- Prefered bigrams: *a bit, a couple, other than, able to, want to, looking forward* and similar ones.
- Frequent use of terms from a social context or describing positive emotions
- Avoidance of *maybe, perhaps* and extensive usage of numbers
- Usage of colloquial phrases, based on verbs, adverbs and pronouns
- Comparably more elaborate replies

Following these principles we implemented basic product responses (greetings, inquiries for product attributes, farewell) for several personalities. All possible replies of our dialogue shell are stored in one XML-file, which we named the *Anthropomorphic Fundamental Base Grammar*. All entries include an associated personality profile, for example:

```
<reply
    query="hello"
    reply="Hello, nice to meet you!"
    ag="1" co="2" ex="1" op="1">
<\reply>
```

[1] http://www.liwc.net/

Which means that this is the greeting of a product with average agreeableness, extraversion and openness and a high value in conscientousness. Another example:

```
<reply
    query="hello"
    reply="Hi! I'm sure I can help you! Just tell me
        what you need and I bet we can figure
        something out!"
    ag="2" co="2" ex="2" op="2">
<\reply>
```

All entries that do not regard any particular personality, should have average personality values in all dimensions.

A central product database with all products and their attributes is extended by the assigned personality profile, i.e. the values in each of the four dimensions. When the application starts up, the dialogue shell retrieves the product data of each product instance and extracts the appropriate entries from the base grammar to build the custom product grammar. If there are no entries that exactly match the given profile, the one that has the most identical values will be chosen. This dialogue shell generates a consistent speech interface to a product by knowing its attributes and a given personality profile, for instance preset by the manufacturer. With the help of such a dialogue shell we plan to fully realize our concept of products with "speaking" personalities.

4 The Digital Sommelier Installation

To test the aforementioned and other previously elaborated concepts, our group developed shopping assistants for two different shopping domains, namely a digital camera store and a grocery store including wines. One shopping assistant provides product information, product comparisons, and personalized advertisements either on tablet PC mounted on the handle of the shopping cart or wall mounted displays. In the case of the grocery scenario one instance of the shopping assistant suggests recipes based on the amount and kind of goods already placed in the shopping cart or provides information about wines and the dishes they go well with.

In a shop the user has the possibility to communicate with the store by means of multi-modal interface as described in [28] or by interacting with the offered products. Her actions in the store are recognized and monitored by a combination of *RFID*- and (in the latest version) by sensor technologies as described in this paper. The shelves in our store are equipped with RFID-readers, and every product is tagged with passive RFID-transponders, which allows for locating and identifying products. Every RFID-transponder also contains a reference to all relevant product information like its name, ingredients and nutrition facts.

In addition to that the DigiSom uses various sensors either directly attached to the wine bottles or in case of champagne to the product box, in order to gather information about the state of a product, i.e. the actual temperature of the bottle of wine and whether the bottle has been shaken or not. Furthermore, interactions with the product box can be monitored using the attached sensors, which are able to measure light, temperature, sound, and acceleration. Whenever the user takes a bottle out of the shelf the attached RFID-tag disappears from the RFID-reader field and the DigiSom receives an event from the EventHeap, which triggers the generation of the according product information page to be displayed on a nearby screen and at the same time initiates the champagne to introduce itself using the personalized voice. Furthermore, the application receives the current temperature of the champagne from the integrated sensor node, comparing the actual temperature against the proposed temperature at which the champagne should be served and generates appropriate hints, e.g. that the champagne has to be chilled before serving (see Figure 5). If the user turns the champagne to read the information on the rear side of the box, the associated sensor detects the rotation and generates the associated event, which is passed to our application. In our application such events are interpreted as an indication of the user's interest in the product and the DigiSom changes the information already displayed. In this case the application refers the user to the web page of the manufacturer, where the user can search for additional information about the product (see Figure 6).

Fig. 5. Generated page with product information including the champagne's actual temperature

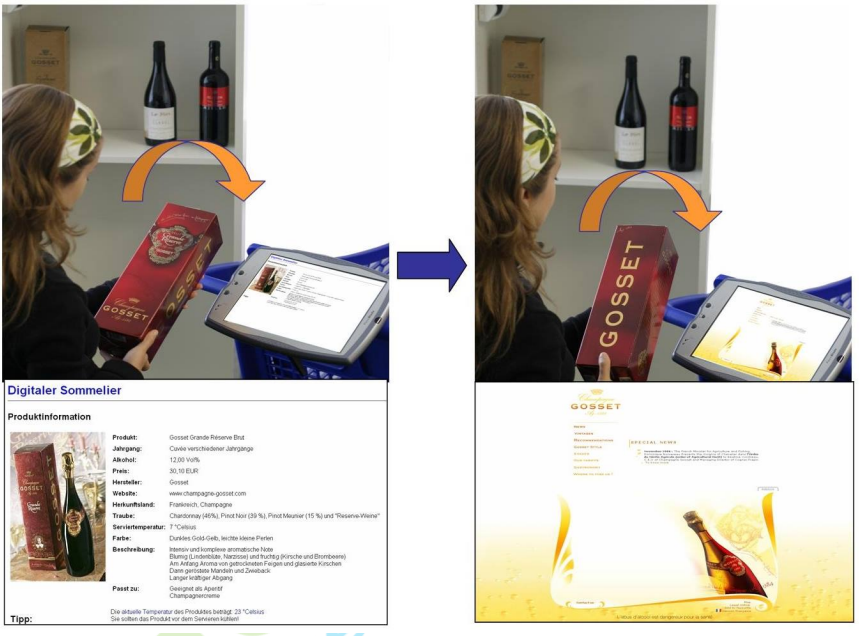


Fig. 6. A user interacting with a bottle of champagne

Through the combination of information from RFID-tags, data about the champagne's/wine's actual temperature, and whether the bottles have been shaken or not, as well as acceleration information from MICAz nodes, our DigiSom is able to incorporate real time data about the state of products to adapt product presentations.

5 Conclusion and Future Work

In this paper we introduced the *Digital Sommelier*, an interactive shopping assistant, which uses the concept of personalized speech to present multi-modal product information. Based on the introduced framework the application combines RFID- and sensor technology to incorporate real time data about the product's state and user interactions with the product itself, in order to adapt the product presentations that will be given to the user. We believe that this technogies will pave their way into new innovative retail solutions. Furthermore, the data collected might also be used to feed and update digital product memories, that can be accessed by producers, retailers and end users. Another promising application fields for anthropomorphic, sensor-based user interfaces are interactive installations in museums or other exhibitions: The areas of education, entertainment and the combination of both provide an appropriate playground for virtual characters and tangible user interfaces, whereas in such an environment the role of integrated sensors would probably focus on detecting user interactions.

We further described a framework for sensor based prototyping, which helped us to quickly develop software components that rely on various sensor data streams. A large set of modules implemented as a Java library allows researchers and developers to assemble applications without the burden of investing much time into low level sensor stream processing. The next step will be to develop a graphical editor, for instance based on the Eclipse Graphical Editing Framework (GEF)[2], to further simplify the process of prototyping. Besides that we will continue integrating new sensor systems that will come to the market.

Our latest research on the interaction metaphor of talking products as presented in this paper will be evaluated in order to discover whether it is possible to communicate brand attributes and intentionally influence the shopping behaviour of customers. It will also be interesting to see, whether this simplified interaction pattern is more intuitive to certain customer groups than other alternatives. Pursuing the anthropomorphic approach we will try to apply an emotional model to products, such that a current emotional state will represent the state of a particular product.

Acknowledgements

This research is being supported by the German Science Foundation (DFG) in its Collaborative Research Center on Resource-Adaptive Cognitive Processes (SFB 378, Project EM 4 BAIR).

References

1. Butz, A., Krüger, A.: A mixed reality room following the generalized peephole metaphor. In: IEEE Computer Graphics & Applications, vol. 26, IEEE press, Los Alamitos (2006)
2. Beigl, M., Decker, C., Krohn, A., Riedel, T., Zimmer, T.: μParts: Low cost sensor networks at scale. In: Adjunct Demo Proceedings of Ubicomp 2005, Tokyo, Japan (2005)
3. Hill, J., Culler, D.: Mica: A wireless platform for deeply embedded networks. IEEE Micro 22, 12–24 (2002)
4. Burge, W.H.: Stream Processing Functions. IBM Journal of Research and Development 19, 12–25 (1975)
5. Stephens, R.: A survey of stream processing. Acta Informatica 34, 491–541 (1997)
6. Johanson, B., Fox, A., Hanrahan, P., Winograd, T.: The event heap: An enabling infrastructure for interactive workspaces. Technical Report CS-2000-02, Stanford University, Stanford, CA (2000)
7. Salber, D., Dey, A.K., Abowd, G.D.: The Context Toolkit: Aiding the Development of Context-Enabled Applications. In: CHI 1999: Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 434–441. ACM Press, New York (1999)
8. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley Longman Publishing Co., Inc., Boston (1995)

[2] http://www.eclipse.org/gef

9. Witten, I.H., Frank, E.: Data Mining: Practical machine learning tools and techniques, 2nd edn. Morgan Kaufmann Publishers Inc., San Francisco (2005)
10. Comption, P., Jansen, R.: A philosophical basis for knowledge acquisition. Knowledge Acquisition 2, 241–257 (1990)
11. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers Inc., San Francisco (1993)
12. Nijholt, A., Rist, T., Tuijnenbreijer, K.: Lost in Ambient Intelligence? Panel Session. In: Proceedings of CHI 2004, ACM, pp. 1725–1726 (2004)
13. Reeves, B., Nass, C.: How people treat computers, television, and new media like real people and places. CSLI Publications and Cambridge university press (1996)
14. Wasinger, R., Krüger, A., Jacobs, O.: Integrating Intra and Extra Gestures into a Mobile and Multimodal Shopping Assistant. In: Gellersen, H.-W., Want, R., Schmidt, A. (eds.) PERVASIVE 2005. LNCS, vol. 3468, pp. 297–314. Springer, Heidelberg (2005)
15. Nass, C., Isbister, K., Lee, E.J.: Truth is beauty: researching embodied conversational agents. Embodied conversational agents, 374–402 (2000)
16. Duggan, B., Deegan, M.: Considerations in the usage of text to speech (tts) in the creation of natural sounding voice enabled web systems. In: ISICT 2003: Proceedings of the 1st international symposium on Information and communication technologies, Trinity College Dublin, pp. 433–438 (2003)
17. Fiske, S., Taylor, S.: Social cognition. McGraw-Hill, New York (1991)
18. North, A., MacKenzie, L., Hargreaves, D.: The effects of musical and voice "'fit"' on responses to advertisements. Journal of Applied Social Psychology 34, 1675–1708 (2004)
19. Aaker, J.: Dimensions of brand personality. Journal of Marketing Research, 342–352 (1997)
20. McRae, R., John, O.: An introduction to the five-factor model and its applications. Journal of Personality 60, 175–215 (1992)
21. Schröder, M., Grice, M.: Expressing vocal effort in concatenative synthesis. In: Proceedings of the 15th International Conference of Phonetic Sciences, pp. 2589–2592 (2003)
22. Schmitz, M., Krüger, A., Schmidt, S.: Modelling personality in voices of talking products through prosodic parameters. In: Proceedings of the 10th International Conference on Intelligent User Interfaces, pp. 313–316 (2007)
23. André, E., Klesen, M., Gebhard, P., Steve Allen, T.R.: Integrating models of personality and emotions into lifelike characters. In: Paiva, A., Martinho, C. (eds.) Proceedings of the workshop on Affect in Interactions - Towards a new Generation of Interfaces in conjunction with the 3rd i3 Annual Conference, Siena, Italy, pp. 136–149 (1999)
24. Costa, P., McCrae, R.: The neo personality inventory manual. Psychological Assessment Resources (1985)
25. Gill, A.J., Oberlander, J., Austin, E.: The perception of e-mail personality at zero-acquaintance. Personality and Individual Differences 40, 497–507 (2006)
26. Pennebaker, J., King, L.: Linguistic styles: Language use as an individual difference. Journal of Personality and Social Psychology 77, 1296–1312 (1999)
27. Nowson, S.: The Language of Weblogs: A study of genre and individual differences. PhD thesis, University of Edinburgh. College of Science and Engineering. School of Informatics (2006)
28. Wasinger, R., Wahlster, W.: The Anthropomorphized Product Shelf: Symmetric Multimodal Interaction with Instrumented Environments. In: Aarts, E., Encarnaço, J.L. (eds.) True Visions: The Emergence of Ambient Intelligence, Springer, Heidelberg (2006)

Socially Intelligent Interfaces for Increased Energy Awareness in the Home

Jussi Karlgren, Lennart E. Fahlén, Anders Wallberg, Pär Hansson, Olov Ståhl, Jonas Söderberg, and Karl-Petter Åkesson

Swedish Institute of Computer Science
Box 1263, 164 29 Kista, Sweden
erg@sics.se

Abstract. This paper describes how home appliances might be enhanced to improve user awareness of energy usage. Households wish to lead comfortable and manageable lives. Balancing this reasonable desire with the environmental and political goal of reducing electricity usage is a challenge that we claim is best met through the design of interfaces that allows users better control of their usage and unobtrusively informs them of the actions of their peers. A set of design principles along these lines is formulated in this paper. We have built a fully functional prototype home appliance with a socially aware interface to signal the aggregate usage of the user's peer group according to these principles, and present the prototype in the paper.

Keywords: smart homes, domestic energy usage, physical programming, connected artifacts, distributed applications, micro level load-balancing

1 Energy Usage in the Home — An Interface Issue

Monitoring and controlling energy usage is a small component in an increasingly intellectually demanding everyday life. Many other choices make demands on the schedule, budget, and attention of the consumer.

However, energy usage is becoming an increasingly important facet of personal choice: the choices made by individual consumers have consequences, both social and environmental. In public discourse, energy usage is discussed as an arena where individual choice makes a real and noticeable difference for the society and the ecological system. This is a driving force and a motivating factor for the individual consumer.

At the same time, another strand of development makes itself known in the home and the household: the incremental digitalisation of home services. The introduction of more and more capable entertainment, media, and gaming appliances on the one hand, and more powerful communication devices on the other, has brought typical homes to a state where there is a high degree of computing power but very little interconnection between appliances. It is easy to predict the future but more difficult to get it right, and the history of "intelligent homes"

C. Floerkemeier et al. (Eds.): IOT 2008, LNCS 4952, pp. 263–275, 2008.
© Springer-Verlag Berlin Heidelberg 2008

and other related research targets is littered with mistaken projections and predictions [1,2]. However, it can safely be envisioned that the future will hold better interconnection between the various computing systems in the home, and that more digital services and appliances for the home will be introduced apace. We believe that systems for energy monitoring and control are an important vehicle for home digitalisation and that they are a natural locus for the convergence of the various functions foreseeable and present in a home.

Our research method is to instantiate various combinations of design principles into fully functional product prototypes, some of which are used to prove a point in a demonstration, others which are field tested, and yet others which are used as a basis for industrial projects. The prototypes are meant to illustrate and test the principles - not to conclusively prove or disprove them.

This paper begins by giving the points of departure for this concrete project, in terms of political and societal goals and in terms of consumer and user needs;it continues by outlining some central design principles taken into consideration during the design cycle; it then describes a concrete prototype for raising energy awareness and for providing unobtrusive mechanisms for better control of electrical energy usage in the home, instantiating the design principles in question.

2 Goals

Current societal and political attention in to a large extent focussed on questions of environmental sustainability. Chief among those question is that of energy usage and turnover. The operation, management, and maintenance of energy usage is to a large extent technological — but based on the interaction between choices made to uphold a desired lifestyle, to conserve energy, participate responsibly in attaining societal objectives, and to preserve the integrity and habitability of one's personal everyday life. To this end, our work recognises a number of background non-technological goals.

2.1 First Goal: Reducing Electricity Usage

An overriding political goal in Europe is to reduce energy usage. While energy is a raw material in several central industrial and infrastructural processes, as well as a key resource in transport and in heating and cooling indoor locations, much of energy usage is incidental to the primary task at hand – that of leading a comfortable life or performing services and producing goods in a comfortable environment. We will in this example focus on one aspect of energy usage, that of electricity usage in household environments.

Household electricity usage has risen noticeably in the most recent ten-year period [3]. There are opportunities to reduce electricity usage, inasmuch much of the increase can be attributed to unaware electricity spill, caused by standby systems or inadvertent power-up of home appliances.

The current public policy on energy matters includes the goal to reverse this trend and to considerably reduce household electricity usage within the next

decade. There are studies that show that in spite of a higher density of kitchen appliances and a rapid and increasing replacement rate as regards such appliances, kitchens use less electricity today than ten years ago[4], which speaks towards the positive effect of technological advancement and towards new technology being a solution, not a hindrance to energy efficiency. Reduction in electric energy usage in households cannot be accomplished through tariff manipulation, since the politically appropriate price range of a unit of electricity cannot provide adequate market differentiation.

2.2 Second Goal: Load Shifting

The marginal cost of electricity at peak load, taking both production and distribution into account is considerable. It is desirable to reduce peak loads by shifting usage from high load to low load times over a 24-hour period or even within a shorter period. Reducing peak loads lessens the risk of overburdening the system, reduces the dependence on marginal electrity production systems – often with more noticeable environmental impact and with greater production cost per unit, and allows the power grid to be specified for a more rational capacity utilisation.

2.3 Third Goal: Preserving the Standard of Living

An immediate political concern is to accomplish a lowered electrical energy turnover and a more balanced load over time, and to do this without impacting negatively on living standards. The savings should be predominantly directed at unconscious spill and waste rather than at reducing quality of life.

In addition to this, the most immediate and pressing need of many users is expressed in terms of life management and coping – we do not wish to add burdens to the harried everyday life of consumers. Our goal is to help the individual household consumer keep electrical energy usage an appropriately small part of everyday life, afford the user higher awareness, better sense of control, without intruding on the general make-up of the home by introducing new, cumbersome, and unaesthetic devices.

3 Studies of Users

Feedback for better awareness or control of energy usage is well studied. A literature review from the University of Oxford [5] indicates savings between 5-15% from direct feedback, and also states that "... time-of-use or real-time pricing may become important as part of more sophisticated load management and as more distributed generation comes on stream.". Much of the work listed in the review concerns feedback in the form of informative billing and various types of displays (even ambient ones), with the aim of providing users with a better understanding of their energy usage. *Smart metering* is a general approach to building better and more intelligent meters to monitor electricity usage, to raise

awareness among consumers, and to provide mechanisms of control either to distributors or consumers: modern electricity meters are frequently built to be sensitive to load balancing issues or tariff variation and can be controlled either through customer configuration or through distributor overrides beyond the control of the consumer [6,7,8]. Better metering has a potential impact on energy load peak reduction and the allows for the possibility for time-of-use pricing, issues which have influenced the design of the tea kettle prototype presented later in this paper.

Furthermore, initiatives such as UK's national Design Councils top policy recommendations from work with users, policy makers, and energy experts, highlight user awareness and control given by e.g. more detailed real-time monitoring of appliance energy usage, controlled through an "allowance", household "energy collaboratives" and energy trading clubs [9].

Our project shares goals with smart metering projects but focusses more on the control aspect than most other approaches — on how to allow users to influence their use of energy in the home environment.

From exploratory interview studies performed in homes and households by ourselves during the prototype design phase of our project, we have found that consumers in general are quite interested in taking control of their energy turnover: they feel they understand little of it and that the configuration and control of their home appliances are less in their hand than the consumers would like them to be. Our subjects in several of the at-home interviews we performed expressed puzzlement and lack of control in face of incomprehensible and information-dense energy bills – none wished to receive more information on the bill itself. For instance, interview subjects were not comfortably aware of the relation between billing units (kWh) and electricity usage in the household.

However, several of our subjects had instituted routines or behaviour to better understand or monitor their energy usage: making periodic notes of the electricity meter figures, admonishing family members to turn off various appliances, limiting the usage of appliances felt to be wasteful of energy, switching energy suppliers to track the least costly rates offered on the market. Many of these routines were ill-informed or ineffectual (e.g. turning off lighting, while not addressing other appliances such as home entertainment systems), and were likely to have little or no effect on the actual electricity usage or cost, but it afforded the users in question some sense of control.

Wishing to gain better control is partially a question of home economics: rising electricity bills motivates consumers to be more aware of electricity usage and to economise. However, while economy alone cannot become a strong enough motivating factor (as noted above, for reasons extraneous to the energy and environment field), consumer behaviour can be guided by other indicators. Societal pressure to behave responsibly accounts for much of consumer behaviour, e.g. in the form of willingness to participate in a more sustainable lifestyle — witness the high rate of return of drink containers in spite of the low remuneration

given by the deposit system. [1] A companion motivator can be found in a general pursuit better to be able to retain the initiative and control over an increasingly complex set of systems in the household.

4 Guiding Principles

To empower users to take control of their household energy turnover, we work to harness the interest consumers show towards the electrical energy usage issue. Our conviction is that to harness this interest we need to design and develop systems that are effortless, fun, and effective to use. We wish to develop pleasurable aspects of new technology — and we do not wish to build systems to use value statements to influence or distress its users. To achieve these goals we suggest a set of guiding principles to follow. These principles are effectivisation, avoiding drastic lifestyle changes, utilizing ambient and physical interfaces, providing comparison mechanism, make systems socially aware and provide both immediate and overview feedback. In the following section we will motivate our choice of principles.

Work towards effectivisation, avoiding drastic lifestyle changes

As discussed in section 2.3, we do not wish to burden the consumer with further tasks or cognitive demands, nor to lower the standard of living in households. Rather then reducing quality of life we need to work towards more effective use of energy in general, and of electric energy in this specific case: effective in both the sense of higher efficiency but also correctly placed in time and space. In doing so we shall not enforce users to introduce drastic life style changes which would become another burden for them.

Use ambient interfaces and physical interfaces

As mentioned above an important aspect of designing artifacts for the home and the household is to avoid drastic life style changes, i.e. not disrupting the behavioural patterns of the inhabitants. Furthermore it is not desired to ruin the aesthetic qualities of the interior. Ambient interfaces [10] share the design principle to not disrupt behavioural patterns and thus the use of ambient interfaces suits very well to be employed. Furthermore utilization of physical interfaces allows us to design interaction that embed into the aesthetics of the artifact and that the actual interaction with the artifact is not drastically changed. [11]

[1] Close to ninety per cent of drink containers are returned by consumers in Sweden, with many European jurisdictions showing similar figures, according to the Swedish Waste Management trade association (www.avfallsverige.se). The presence of a deposit system seems to have a strong effect on the willingness to make the effort to return containers, even while the deposit usually is low and does not translate into adequate remuneration for the time and effort spent.

Use comparison mechanisms

To reduce energy usage we must slowly change the user's behaviours. As a basis for any system for behavioural modification, we must provide a benchmark or measure with which to compare individual (in our case, typically aggregated by household) performance. We will want to provide comparison mechanisms to give the consumer a sense of change, both as regards overall lifestyle and for individual actions which influence energy turnover in the home. In our current set of prototypes, we explore comparison over time ("Am I doing better than last week") and over peer groups (using a recommendation system framework [12,13]).

Build socially aware systems

Energy usage and related behaviour, as intimated above, is not only an individual question. Use aggregated over a large number of households will have greater impact both on general usage levels and on load balance than will individual usage profiles; in addition, the effects of social pressure, both to conform and to lead, can be harnessed to make a difference. Therefore it is important to create designs that operationalise some of the social context that normally is less visible in household situations. If this is done in a manner which avoids pitting users against each other in individual contests of appropriate behaviour, and which does not succumb to pointing fingers at those who do not conform to norms, such a social aware system will provide users to not only compare or learn from their own usage but others as well.

Immediate feedback and overview feedback

While the learning which is required by the consumer is on a fairly high cognitive level, and is intended to increase awareness and contribute to a sense of empowerment and control, it should be based on standard learning mechanisms. It is well established that in any learning situation, the timing of the stimulus must be in relevant juxtaposition with the contigent action — the highlevel behaviour or individual actions the consumer is engaged in. [14] The stimuli or signals to provide as feedback to the user must be *appropriate* and have *informational value* — in our case, they must not be overloaded with other information and not clash with other signals or messages the user may be interested in, and it must also have relevant cognitive content so as not to deteriorate into mere background noise. Specifically, we will keep separate the immediate feedback necessary to learn from actions from the overview sense necessary to be aware of lifestyle effects.

Plan and build for added benefits

The economical benefits provided through higher energy awareness is not enough to catch the eye of consumers, but must include other benefits and thus be designed as part of a larger platform. One possible added benefit is to provide

added control and to empower the indvidual consumer and emergent groups of consumers. A possible integration of energy issues with e.g. safety monitors, time management services, and communication systems might be one solution.

5 Prototype Example – The Socially Aware Tea Kettle

Electric stand-alone tea kettles are the preferred device for heating water if the alternative is using a pot on an electric stove: tea kettles are much more energy efficient. To contribute to the overriding goal of reducing electricity usage it is thus useful to encourage households to move from stove-top saucepans to stand-alone kettles.

However, kettles occasion usage peaks: they use power up to 2 kW and are among those home appliances which require the highest current when switched on. This is observable by the consumer, in that switching on a tea kettle for many consumers causes a minor but visible brownout in the home, dimming lights momentarily. In addition, kettle usage patterns are cyclic. People have more or less the same diurnal behaviour patterns and tend to heat water for hot beverages and for other cooking purposes at about the same times.

The tea kettle is thus a useful and illustrative example of household electricity appliance: it has a high wattage and its usage is non-random with similar usage patterns across households.

As noted above, reducing peak loads by shifting usage is a desirable goal. To this end, to illustrate the possibilities inherent in aggregating households into cooperating pools of users for load-balancing purposes, we have built a socially aware tea kettle, suitable for pooling usage across several households. Pooling usage and reducing peak loads allows an aggregation of households the potential to negotiate lower average rates (possibly offset by higher peak load rates at the margin). In keeping with the guiding principles given above, we do not wish to burden the consumer with calculations or other cognitively demanding operations at the point in time where their focus is on producing hot beverages. Instead of providing numerical or graphical information, deflecting the attention of the consumer from tea to tariffs and time schedules, we provide an enhanced tool whose primary purpose remains heating water.

The underlying assumption is that if a number of households with suitably dissimilar habits are pooled, their power requirement can be balanced over a time period, if less time-critical energy requirements at times can be shifted from the immediate moment they are ordered to some less loaded point in time. A typical example of pooling currently under experimental deployment in households is delaying hot-water cisterns from immediately replacing the hot water used by morning baths and showers. If hot-water cisterns are programmed to heat the water when the overall network load is low, the aggregate cost of heating water can be kept down [7,8]. Figure 1 illustrates how the power usage of various electrical appliances (examplied by tea kettles) can be spread out in time to acheive a better load balance if the households are pooled.

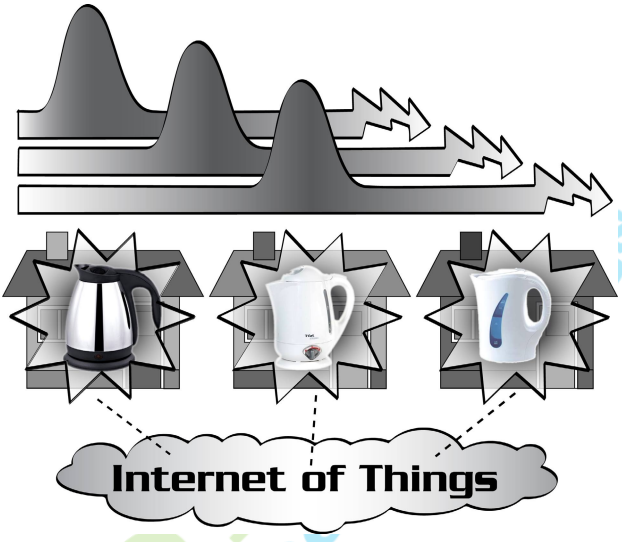


Fig. 1. Pooling usage to achieve load balance

Similarly, we posit that heating water is only partially time-critical: when someone orders up a pot of boiling water, quite frequently the hot water can be delivered within a flexible time window rather than immediately. If a kettle can be made aware of the current and near-future load among a number of pooled households, it can inform its user of when within the next few minutes a kettleful water can be heated at lowest strain to the delivery grid (and, presumably, at lowest cost, if price is used as an additional signal to consumers).

Our example fully-functional kettle is mounted on a rotating base, and provides the user with rotational force feed-back to indicate booked capacity in a window over the next few minutes. A picture of the current prototype is shown in Figure 2. This allows the user to rotate the kettle to get a sense of when the load is light (less force is needed) or heavy (the kettle resists being rotated into that position). The user is always able to override the recommended time slot, and select immediate delivery or non-peak time.

5.1 Technology

The wooden cabinet on top of which the tea kettle is placed contains various hardware devices that are necessary for its operation. At the heart of this setup is a Java microcontroller responsible for Internet communication as well as measurement and control functionality. Using a clamp meter connected through i2c interface, the microcontroller monitors electric current used by the kettle. An i2c connected 240V power switch is incorporated to allow the processor to start and stop the tea kettle at designated times. The prototype also uses a modified force feedback steering wheel, modified from an off-the-shelf car game controller,

Fig. 2. Current kettle prototype. A production unit will have the same form factor as a standard off the shelf tea kettle.

connected to the base of the kettle to yield the friction-based rotation of the kettle. In addition, there is an array of LEDs positioned around the base of the kettle used to visually convey information such as active bookings.

The microcontroller runs a kettle software process which controls the hardware devices inside the wooden cabinet (forced feedback device, power switch, etc.), and also communicates with a household booking service over IP. The booking service is connected to all household electrical appliances, and monitors their current as well as predicted future power usage. The booking service is in turn connected and reports to a pooling service, which supervises the activity over many households. The pooling service is thus able to construct an aggregated "profile" of the predicted near-future power consumption of all connected households. This profile is then continuously communicated back to the booking services, and finally the individual appliances (e.g., the tea kettle prototype), where it is used to convey operating conditions that the user may wish to consider. The profile may be biased by any changes in the power suppliers future cost levels or by maximum power usage limits that may be stipulated in contracts between the power supplier and the pooled households. In the tea kettle prototype the profile is used to control the force feed-back mechanism acting on the kettle. The friction at a certain angle of rotation represents the predicted load a number of minutes into the future. The system architecture is illustrated in Figure 3. The kettle software control process, as well as the booking and pooling services, are developed using PART [15]. PART is a light-weight middleware,

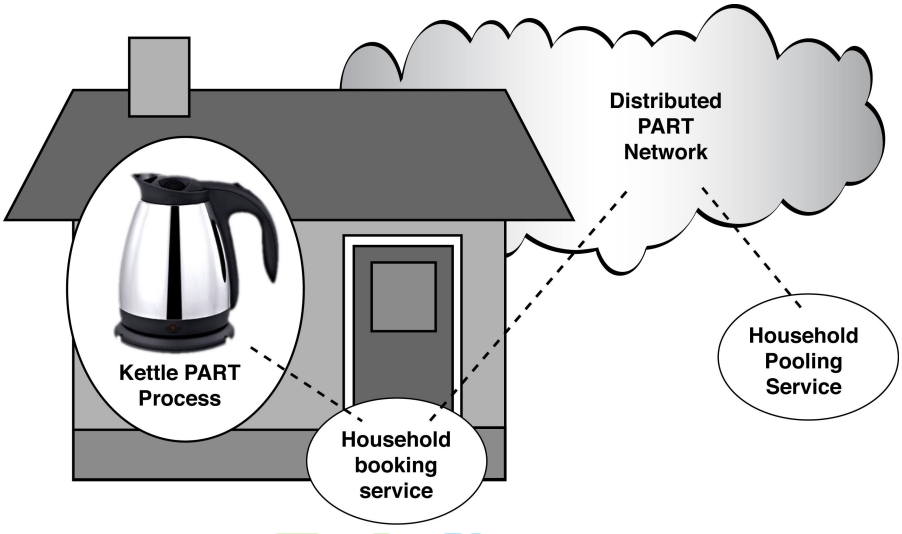


Fig. 3. System overview

written in Java, for developing pervasive applications that run on a range of different devices. PART is available as open source, under BSD license, and can be downloaded at http://part.sf.net.

The kettle can be rotated approximately 360 degrees, which currently is set to correspond to fifteen minutes of time. At one end of the interval (starting from degree 0), the friction feedback to the user indicates bookings and thus the predicted load in the immediate near future, while at the other end (around degree 360), the user can sense the predicted load fifteen minutes further along in time. When the user has found a suitable position, the kettle is activated by pressing the ordinary "on" button. This event is caught by the kettle controller hardware, which will temporarily switch off the kettle's power. The power will then be switched back on by the controller at the time corresponding to the kettle rotation, which will cause the water to start heating.

The reason why the rotation interval is fairly short (minutes rather than hours) is that we believe that roughly fifteen minutes is as long as tea and coffee drinkers can be expected to wait for their hot water. The assumption is that users want their hot water now, but that they might be prepared to wait a while in order to getter a better price, or to help spread the load among their peers. We have considered it unlikely that the user would want to schedule the heating of water for cooking or hot beverages hours into the future. One implication of this is that we also believe that once the user starts to turn the kettle, the chances are high that a booking will actually be made (the user will want the hot water now!), even if the load turns out to be high in the entire fifteen minute interval. This has allowed us to keep the interface fairly simple, based more or less entirely on the force feedback mechanism, and avoiding displays, buttons, etc,

which might would have been required to support a more complex booking procedure (e.g., allowing the user to set constraints on price, time, load, etc.) All of these assumptions have to be confirmed via user studies, which are planned for the near future.

5.2 The Home Configuration Panel

Once the previous prototype is in place, the integration of various home services will hinge on getting an overview of the home configuration. How this can be achieved without overwhelming the user is a daunting design task, and this interface is only in planning stages as of yet – but it will involve a careful combination of well designed controls with defaults, examples, and preferential selections in face of difficult choice. [16,17]

We foresee in this case an integration of energy issues with e.g. safety monitors, time management services, and communication systems.

5.3 Community Tool

Similarly to the home configuration panel, an overview and control system is needed for the pooled community household power requirements. This could involve social awareness mechanisms as well as an opportunity to test different business models towards the provider. Current implementation consists of a graph visualizing web browser applet communicating to the rest of the system via a servlet.

6 Conclusions

This prototype is designed to illustrate three strands of thought. It works towards an increased *social awareness* in the household; it is based on an informed design of *immediate feedback* of aggregate information; it is an example of *unobtrusive design* to fit in the household without marring its home-like qualities, and as a platform and base for further design work, it combines our design principles in a fully functional prototype.

The kettle is an instantiation of our design principles, but by no means the only possible one, and is not intended to be deployed in the consumer market as is – it serves as an illustration of how social factors are or can be a parameter in our behaviour even within our homes. The underlying principle of an internet of household appliances can be instantiated in many ways, of which this is but one: energy usage is not a prototypically social activity, but in aggregating the behaviour of many peers into a pool of usage it becomes more social than before. The point of this prototype is that with a tangible yet unobtrusive interface we are able to provide users with a sense of social context and superimpose a behavioural overlay over individual action to aggregate usage patterns into a whole which is more functional from a societal perspective. We will be exploring this avenue of social technology in further prototypes, with other design realisations of the same principles.

Our tea kettle prototype illustrates the importance of load balancing and of user control. However, it also raises the requirement of providing users with immediate feedback directly connected to action – a well-established principle in the study of learning. As further developments of that principle we are field testing variants of electricity meters which provide both overview information of aggregated weekly usage and in other instances reacts to user action by giving auditory and visible feedback when an appliance is connected to the net.

Our tea kettle prototype illustrates the possibility of designing an interface which both provides immediate lean-forward information for the action the user is about to engage in, but without requiring cognitive effort – the information is encoded in rotation and visual feedback which needs little or no parsing. The attention of the user is not deflected from the primary goal at hand; the tool remains a tool designed for its primary function.

Acknowledgements

The research reported in this paper is part of and sponsored by the Swedish Energy Agency research program "IT, design and energy", project ERG, project number 30150-1. The ERG project adresses consumer awareness, understanding, and control of the energy usage of tomorrow's household.

References

1. Harper, R.: Inside the smart home: Ideas, possibilities and methods. In: Harper, R. (ed.) Inside the Smart Home: Interdisciplinary perspectives on the design and shaping of domestic computing, pp. 1–7. Springer, London (2003)
2. Cook, D.J., Das, S.K.: How smart are our environments? an updated look at the state of the art. Pervasive Mobible Computing 3, 53–73 (2007)
3. Energy Indicators. Annual Report from The Swedish Energy Agency (2005)
4. Bladh, M.: Snålare apparater men ökad elanvändning. In: Vardagens elvanor. pp. 18–20. (in Swedish)
5. Darby, S.: The Effectiveness of Feedback on Energy Consumption. University of Oxford, Oxford (April 2006)
6. Owen, G., Ward, J.: Smart meters: commercial, policy and regulatory drivers. Sustainability First, London (2006)
7. Abaravicius, J., Pyrko, J.: Load management from an environmental perspective. Energy & Environment 17 (2006)
8. Abaravicius, J., Sernhed, K., Pyrko, J.: More or less about data-analyzing load demand in residential houses. In: ACEEE 2006 Summer Study, Pacific Grove, California (2007)
9. Design Council Policy Paper: Designing for a changing climate. The UK Design Council (December 2007)
10. Wisneski, C., Ishii, H., Dahley, A., Gorbet, M., Brave, S., Ullmer, B., Yarin, P.: Ambient Displays: Turning Architectural Space into an Interface between People and Digital Information. In: Streitz, N.A., Konomi, S., Burkhardt, H.-J. (eds.) CoBuild 1998. LNCS, vol. 1370, Springer, Heidelberg (1998)

11. Ullmer, B., Ishii, H.: Emerging frameworks for tangible user interfaces. IBM Systems Journal, 915–931 (2000)
12. Karlgren, J.: Newsgroup clustering based on user behavior — a recommendation algebra. In: TR T94004, SICS, Stockholm, Sweden (February 1994)
13. Höök, K., Munro, A., Benyon, D.: Designing Information Spaces: The Social Navigation Approach. Springer, Heidelberg (2002)
14. Kirsch, I., Lynn, S.J., Vigorito, M., Miller, R.R.: The role of cognition in classical and operant conditioning. Journal of Clinical Psychology 60, 369–392 (2004)
15. PART Manual: PART-Pervasive Applications Runtime (November 2006) (retrieved September 19 (2007)
16. Humble, J., Crabtree, A., Hemmings, T., Åkesson, K.P., Koleva, B., Rodden, T., Hansson, P.: Playing with the bits - user-configuration of ubiquitous domestic environments. In: Dey, A.K., Schmidt, A., McCarthy, J.F. (eds.) UbiComp 2003. LNCS, vol. 2864, Springer, Heidelberg (2003)
17. Rodden, T., Crabtree, A., Hemmings, T., Koleva, B., Humble, J., Åkesson, K.P., Hansson, P.: Assembling connected cooperative residential domains. In: Streitz, N.A., Kameas, A.D., Mavrommati, I. (eds.) The Disappearing Computer. LNCS, vol. 4500, pp. 120–142. Springer, Heidelberg (2007)

IOT for All
Internet of Things

Connect with Things through Instant Messaging

Jongmyung Choi[1] and Chae-Woo Yoo[2]

[1] Dept. of Computer Engineering, Mokpo National University,
61 Cheonggye, Muan-gun, Jeonnam 534-729, Korea
jmchoi@mokpo.ac.kr
[2] School of Computing, Soongsil University,
Sangdodong, Dongjakgu, Seoul 156-743, Korea
cwyoo@ssu.ac.kr

Abstract. As society and technology advance, we need more frequent communication not only with coworkers but also with "things." In this paper, we introduce a new communication paradigm and a prototype system that implements the paradigm. It is a communication façade that supports human-human, human-thing, and thing-thing communication using instant messaging. With this system, we can add any things worth talking with to our buddy list, and communicate with it through its instant messaging agent that encapsulates its services and status. We adopt a pipe & filter architecture pattern for the agent due to the requirements of extensibility. For efficient cooperation, presence information is also needed, and we suggest a statechart based presence model of things.

Keywords. Instant Messaging, Human-Thing Communication, Thing-Thing Communication.

1 Introduction

As society and technology become more complicated, we need more frequent human-human, human-thing, and thing-thing communication and cooperation. Human-human communication and cooperation have progressed using the Web and instant messaging. The Web allows people to share knowledge asynchronously and to build social networks and collective intelligence over the Internet, and instant messaging [1,2,4,7] makes it possible for people at a distance to communicate and cooperate in real time.

Compared to human-human cooperation, there are significant difficulties in human-thing and thing-thing communication and cooperation. First, there are no distinguishing technologies or paradigms for these kinds of communication and cooperation. Second, there are no standards for human-thing and thing-thing interaction. Third, there are no killer applications that support human-thing and thing-thing communication and cooperation.

Many researchers have been working to address these problems, and some have tried to extend instant messaging to connect people with devices and applications [3,10,11,12,14]. Because this approach is simple and has many advantages, we adopt it in this paper, and extend the concept of the entity in communication to include any entity worth being added, and the agent for the entity is also added. Therefore we can

C. Floerkemeier et al. (Eds.): IOT 2008, LNCS 4952, pp. 276–288, 2008.
© Springer-Verlag Berlin Heidelberg 2008

add any things to our buddy list of instant messaging, and communicate with them. For example, we can add a conference room to our buddy list, and we can talk to the room to reserve it for our project meeting. Furthermore, I can ask my printer to print my project report, and my printer can talk to me when it runs out of paper.

In this paper, we propose the concept of unified instant messaging as a communication façade, and introduce some scenarios for human-thing and thing-thing communication. For these communications, we introduce the architecture of smart agents for things. The agent consists of client module, interpreter module, and external interface module. We also suggest a presence model for things. It can be modeled with a statechart and tree.

In this paper, our contributions are three:

- Introduce a new paradigm for human-thing and thing-thing communication and cooperation
- Propose a pipe & filter type architecture for an instant messaging agent
- Propose a presence model for things

Our work will break down the barrier between human and thing in communication cooperation. It will also increase ease of work both in the workplace and at home with the support of devices, services, and things.

The remainder of the paper is organized as follows. In Section 2, we discuss other works that are closely related to our work. Then we propose our instant messaging system that supports human-thing and thing-thing cooperation and communication in Section 3. After that, in Section 4 we introduce our prototype system. Finally, we reveal the conclusions of our work in Section 5.

2 Related Works

Most of the work on instant messaging may be classified into two groups. Projects in the first group are focused on the influence of instant messaging on our daily life. Therefore, most of the research topics address how many users are using instant messaging [1], the main features in its use [4,5], why users are using it [4], how users use it in their work place [1,4,5], how teenagers use it for social networking [6], etc. These studies are very important for identifying co-working patterns with instant messaging, identifying new requirements for the workplace, etc.

Projects in the second group focus on extending instant messaging by adding new features [8-13]. Some of these projects are similar to our work. Adams [13] introduces a mail notification application based on instant messaging. It is a kind of human-application communication. Arjun [10] introduces a usage scenario in which instant messaging allows communication between networked appliances. In addition, Simon Aurell [11] shows how instant messaging can be used for monitoring and controlling remote devices in home automation. It is a kind of human-device communication. In this work, the intelligent gateway is the main part of the system, and it communicates with devices using device drivers and some form of translators. Marcela Rodríguez [14,15,16] also introduces location-aware instant messaging that supports communication with devices. Cian Foley [12] extends the concept of communication further, and introduces instant messaging that provides person-to-person, person-to-device, and person-to-service communication. Instant messaging allows the addition of services and devices to the buddy lists, and to access them in a uniform way.

Our work is differentiated from other works in three ways. First, we extend the communication entity to anything that is worthwhile talking with. Therefore, we use instant messaging as a uniform platform for communication and cooperation with people, devices, services, virtual objects, and things. Second, we introduce an extensible architecture for communication and cooperation with things. It has filter type architecture, and developers can add more features without rebuilding the systems. Third, we propose a presence model for things based on a statechart.

3 Communication Façade

3.1 Requirements for Communication Façade

In the near future, communication and cooperation between people and things or machines will be more prevalent, and its importance will increase. In the digital appliance field, there is already an identified need for human-machine and machine-machine communication in building smart homes. For example, a smart fridge [17] connects with the Internet, and it has the capacity to check its contents and send a shopping list to the owner's cell phone through short message services (SMS) or directly to the supermarket. It means that the fridge tries to communicate with humans and with things. However, we still have problems in the interaction between human and machine or between machine and machine because we do not have a unified communication method and protocol. As a result, we have difficulties due to the complex communication paths between humans and things as illustrated in Fig. 1. Furthermore, a path may be implemented with different vendor-specific protocols.

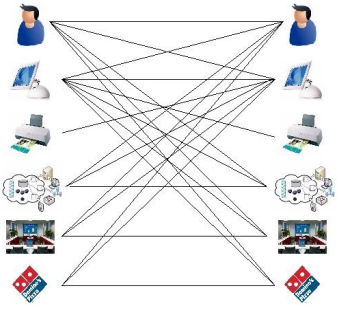


Fig. 1. Complex Communication Paths

To boost and support human-thing and thing-thing communication, we need a unified and standardized communication method. If application developers and appliance manufacturers use their own proprietary communication methods, users will experience confusion and difficulty in using the systems. Therefore, a standard way of communication is needed. For this, we have four requirements to meet. First, it must simultaneously be both synchronous and asynchronous. Because the style of the communication is real-time exchange of information or requests for services, it must be synchronous. And, sometimes it is asynchronous because it supports asynchronous communication for unavailable things.

Second, the communication method must be simple enough for a machine to understand and easy enough for users to use in comfort. A great deal of work has been done in developing convenient communication methods between humans and machines, and these methods include voice, motion, embedded menu, text, etc. Voice and motion recognition methods are very convenient for users, but they require extensive resources and there are difficulties in implementing this kind of system. On the other hand, embedded menu is physically limited, and it cannot be accessed at a distance.

Third, the communication must be extended over the network or the Internet. Then we can communicate with remote things, and things can cooperate with each other over the network. For thing-thing cooperation, the network connection is inevitable. Some intelligent agents can be in between the things and the network, and they translate the communication messages for both side of communication.

Fourth, the communication method must be easily extensible for two reasons. First, it must be able to adopt a new kind of thing or machine with the least efforts. Second, it must be able to adopt more intelligence without rebuilding the system. To support the extensibility, the communication method must be structured and encapsulated.

For a unified and standardized communication method, instant messaging is a promising technology. Fig. 2 shows that instant messaging plays the role of the communication façade for human-human, human-thing, and thing-thing communication. By using instant messaging as a gateway, we can reduce the complexity of multiple communication paths and vendor-specific protocols.

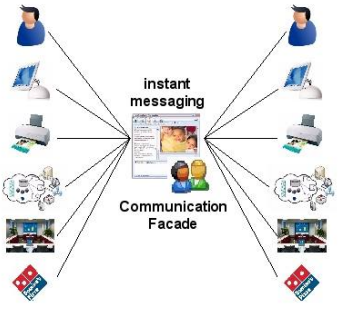


Fig. 2. Communication Facade

Instant messaging meets the four requirements mentioned above. First, it supports synchronous and asynchronous communication, and it is well suited to real time cooperation. Second, it is based on text, so it is rather simple to implement compared to voice or motion. Furthermore it is well-known to be easy to use as shown by its popularity. Third, it is connected to the Internet, and any user at a distance can communicate and cooperate with things. Fourth, instant messaging can be extended rather easily.

In instant messaging, the communication facade is possible because of virtual counterparts, called instant messaging agents of physical things. Fig. 3 illustrates the relationship of physical things with their virtual counterparts, or agents. An agent encapsulates the state and behavior of a physical thing. Fig. 3 also shows some examples of agents. With the concept of agent, we can communicate with computers, devices, services, things, or even stores in the same way as we do with our co-workers.

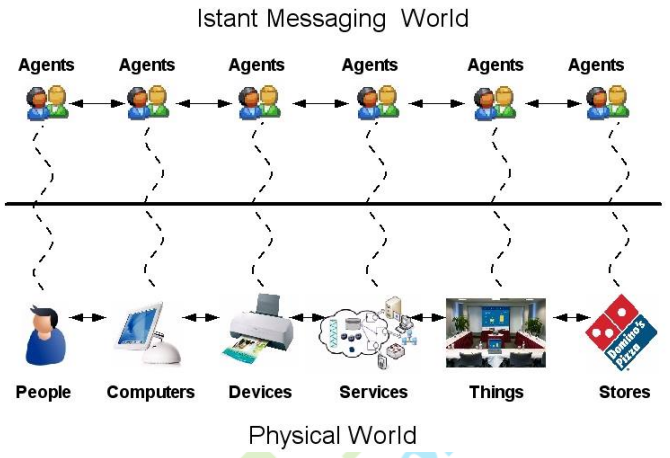


Fig. 3. Connection Physical World and Instant Messaging World

Besides the meeting the four requirements, instant messaging has one more significant advantage. By using instant messaging, we can utilize the existing instant messaging systems' facilities for communication and cooperation. For example, users can connect with things from mobile phone or PDA through CDMA or GSM networks without extra effort because these facilities are already established. Furthermore, users can also enjoy the voice interface supporting instant messaging [2].

3.2 Communication Scenario

Instant messaging removes the barrier between humans and things, and it allows people to communicate with computers, devices, services, things, etc by providing a simple and unified method. If we can communicate with devices, we can work more efficiently and easily. Here is a scenario that shows how to communicate by better utilizing our daily devices.

Scenario 1

A project manager, Choi, is far away from his office, and he notices that he needs to print his presentation handouts before the project meeting. He pulls out his mobile phone, and he selects "desktop" from the buddy lists of instant messaging on his phone. After the selection, a communication window opens to talk with the "desktop," and he then sends a message to print his presentation data before he comes to the office. A few seconds later, he gets a message from his "printer" through instant messaging, and the message is that it has run out of paper. He searches available coworkers in his office by checking the presence and availability marks in his instant messaging, and he asks one of them to fill his printer with paper. About five minutes later, his printer sends a message that it has finished the job.

Scenario 1 shows communication between human, computer, and printer. The real communication follows the sequence diagram illustrated in Fig. 4. In human-thing

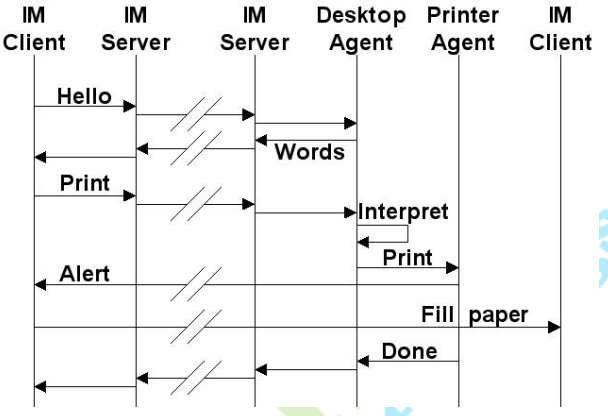


Fig. 4. Sequence Diagram for Scenario 1

communication, it starts with the "Hello" message. On receiving the "Hello" message, the thing agent sends its vocabularies and syntax for further communication.

Humans can communicate with things with instant messaging. For example, we can communicate with a conference room to reserve the room for the project meeting. Here is a scenario of communication with the room.

Scenario 2

Choi needs to reserve a conference room for the project meeting. He asks a conference room called "Ocean," whether it is available at 3:00 p.m., and it replies that it is available until 5:00 p.m. He responds to it and reserves the room from 3:00 to 5:00. The project team members see that "Ocean" is marked in their instant messaging clients with a message about the project meeting.

Fig. 5 shows the sequence diagram for scenario 2. Room scheduler is an existing application for managing and scheduling rooms, and the room agent contacts it for schedule information.

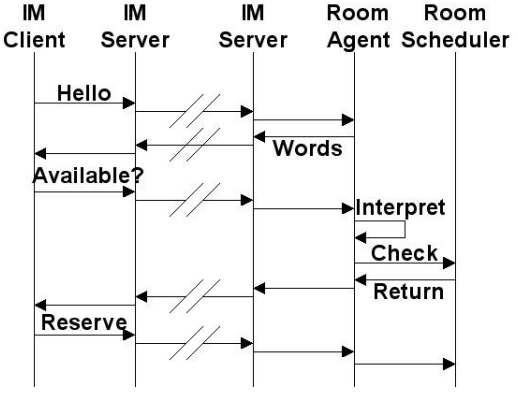


Fig. 5. Sequence Diagram for Scenario 2

Things can communicate and cooperate with other things using instant messaging. However, at least one of the communication and cooperation agents must have computing power. Here is a scenario for cooperation.

Scenario 3
After the reservation, Choi asks "Ocean" to prepare for the meeting. The room checks the availability of projector and computer, and talks to the computer to get the presentation file from Choi. Then the computer talks to him to get the file, and he sends the file to the computer. When everything for the meeting is prepared, the room talks him that the meeting room is ready.

Fig. 6 shows the sequence diagram for the scenario 3. The room agent must have the ability to check for the presence of a computer and projector in the room and the service availability of them.

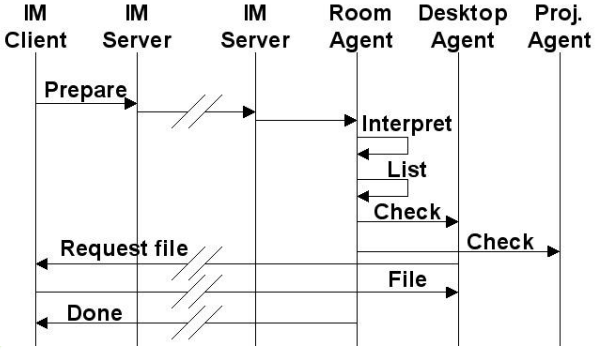


Fig. 6. Sequence Diagram for Scenario 3

3.3 System Architecture

An instant messaging system consists of servers and clients, and it follows peer-to-peer system architecture [19]. The main roles of the server are transferring messages and storing messages temporarily for clients. The message format follows Extensible Messaging and Presence Protocol (XMPP) [21]. An instant messaging client has direct connection with people or things, and we call it an instant messaging agent. Fig. 7 shows the architecture of instant messaging systems. In LAN, people use an instant messaging agent that runs on desktop computers, and they can connect with devices, services, and things through the instant messaging server. When people are outside, they can use a mobile instant messaging client that runs on mobile phone or PDA.

The instant messaging agent plays the central role in the communication façade because it enables people to connect with things. It consists of client module, interpreter module, and an external interface module. Fig. 8 shows an agent's architecture for a printer device.

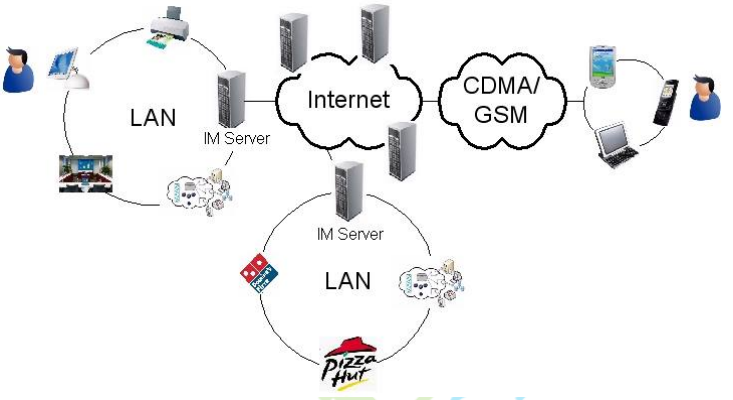


Fig. 7. Architecture of Instant Messaging System

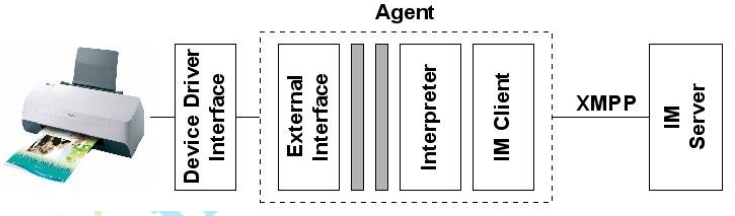


Fig. 8. Instant Messaging Agent's Architecture

In Figure 8, the client module is the most basic part for communication. It takes charge of the communication with an instant messaging server over the TCP/IP sockets, and it also manages the communication sessions. Another role is to parse the message in XML format (XMPP), extract the core message - we call it thing-specific message - from the whole message, and deliver it to the interpreter module.

Interpreter module is the brain of the instant messaging agent. It receives a thing-specific message from the client module. The message is the core of the data, and its format may be in application-specific XML format. For example, the message to a printer device is regarding sending requests to a printer or about getting information about the printer. Therefore, the interpreter module parses the thing-specific message transferred from the client module, understands the meaning of it, and translates it to the devices' commands or requests to devices. Most intelligent services are processed within this module.

External interface module is for connection with external things. The roles of this module are classified into two parts. First, it sends requests to things or devices, and the requests must be understood by the things. Second, it receives responses of requests and the presence information from the things. The response and status messages must be transformed into general terms in this module for corresponding parties.

The agent follows the pipe & filter pattern [20]. We can add more modules between the interpreter and the external interface module without recompiling the source. The filter module may support the interpreter module for more intelligent services.

3.4 Presence Model

Presence [18] is a very important feature in instant messaging, because it provides the presence information of other users. In communication façade, an instant messaging agent may have a variety of status information, because the agent may represent various things such as people, printer, room, etc.

In communication façade, the type of presence is classified into two: physical and logical. Physical presence means that a thing locates in a specific range so that it is available physically. Logical presence means the status of service availability. In communication façade, the importance of physical presence will increase. For example, in scenario 3, the room agent has to check the physical presence information of projector and computer. The physical presence information server can be implemented with location systems such as GPS or RFID [23]. Physical presence has one of two values: exist or non-exist.

The physical presence information of things is monitored and is provided by the presence monitor. Fig. 9 illustrates the architecture of presence server. The presence monitor checks the presence of things and their location information. If a thing is out of its location, the presence monitor reports that its physical presence is "non-exist."

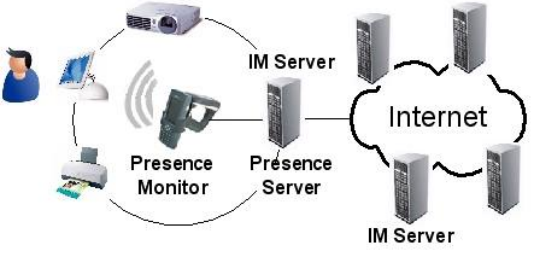


Fig. 9. Physical Presence Server Architecture

In contrast to physical presence, logical presence may have one of a variety of values. With regard to value, logical presence information has two features. First, some of the status values of presence are exclusive to each other. In this case, we say that the status values are in equal level. Second, the other status values are contained in another value. For example, an agent for a printer device can have off-line, on-line, printing, out-of-paper, and paper-jam. Then off-line and on-line are exclusive. However, printing, out-of-paper, and paper-jam are contained in on-line, and each of them is exclusive.

Logical presence is naturally modeled with UML statechart [22], because the status change is well modeled with statechart. All states in equal level have a transition of value among them, and the contained states are represented with internal states. Fig. 10 shows an example of a statechart for printer agent.

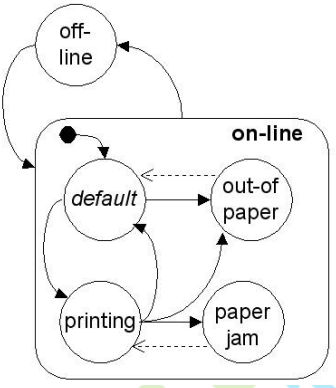


Fig. 10. Presence State Diagram

The statechart model can be transformed into a tree model. In the tree model, the root node is the name of a thing, and its child nodes are non-contained exclusive values. In addition, a node and its child nodes are determined by the containment of status values. The container is parent node and the containees are child nodes. Fig. 11 shows the tree model for a printer model.

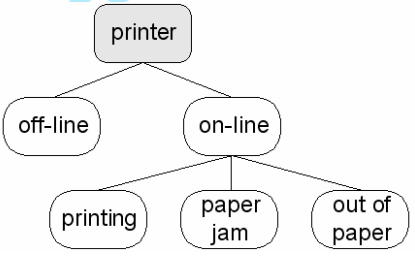


Fig. 11. Presence State Tree

4 Prototype System

We built first version of prototype system for communication façade paradigm. The main goal of this prototype is to implement scenario 1 and 2. We utilized Jabber [24] and XMPP protocol for implementing the system. Jabber is an open instant messaging and presence awareness system. This server is used to notify the state of people and agents, and to handle the interaction between people, agents and devices through XML messages. We used the existing instant messaging server for communication server, and we developed instant messaging agents with the Java language, especially with JDK 1.6 on Win 32 platform. Development of our prototype system was fairly straightforward because it utilized existing open source systems such as Jabber, and java shell [25].

The architecture of our prototype system is illustrated in Fig. 12. Our prototype system consists of five agents: Printer agent, Desktop agent, Room agent, Store agent, and BBS agent. The room agent uses database modules to check for room reservation information and to reserve a room by querying the database. Store agent uses order module to order

goods, but we deployed it in another computer in our lab instead of real store for testing. Order System is so simple that it prints out order information to console, because it is developed just for testing. Desktop agent provides the ability to talk with a desktop computer, and it is based on an open source java shell [25]. It is able to execute commands by creating new processes. Printer agent has ability to print documents and to notify its status to users. To connect with a device driver, it uses JNI module.

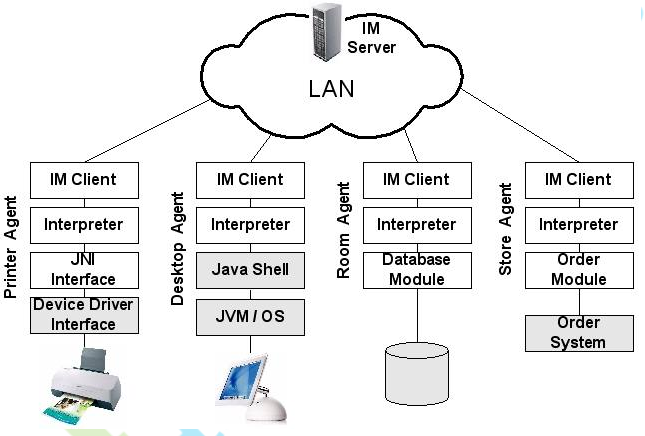


Fig. 12. Architecture of Prototype System

Out prototype system supports communication with co-workers, devices, applications, and things. Fig. 13 shows a snapshot of communication with a printer. When the printer runs out of paper, it notifies users about its status.

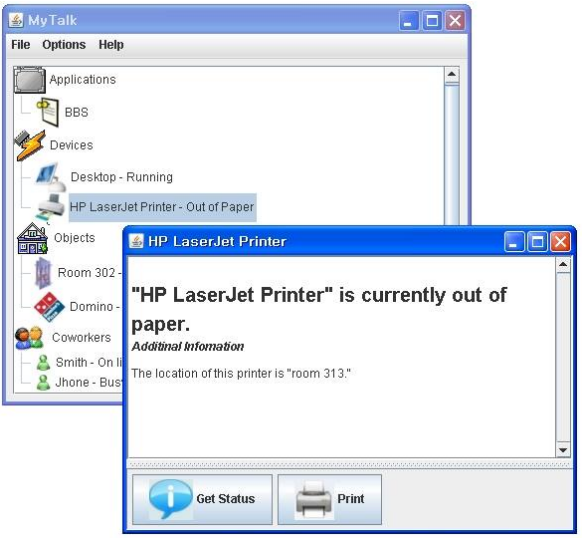


Fig. 13. Snapshot of Communication

Currently, we are developing the second version by extending the first prototype system to support scenario 3 and fix some problems. We re-implement Desktop agent and Printer agent with Microsoft Foundation Classes (MFC) library for more convenient control of the desktop computer and the printer. For supporting scenario 3, we build a presence server. For physical presence service, we use a 900M Hz RFID reader. The reader is able to check the physical presence of a beam projector and a notebook computer within the conference room by reading tag identifiers attached to them.

5 Conclusions

As technologies advance, we need more convenient and efficient methods for communication and cooperation with other people and even with things. There has been a great deal of work done for convenient human-thing and thing-thing communications, but no great success until now.

In this paper, we suggested a new paradigm for human-thing and thing-thing communication and cooperation by extending existing instant messaging. Instant messaging supports text-based real-time communication and cooperation. In addition, many additional features such as running on mobile phone or supporting voce communication are developed. The most important feature of instant messaging is its extensibility. We can add new features for human-thing and thing-thing communication façade with little efforts by adding messaging agents to the existing instant messaging systems.

We suggest instant messaging agent that enables things to talk and cooperate with people. An agent represents a thing or a device, and it keeps the information and status of the thing. Sometimes, it interprets the requests from users and executes the device on behalf of the users. Its architecture follows the pipe & filter pattern, and developers can add more features without rebuilding the system. It provide the presence model information for things, and for this, we suggested a presence model based on the statechart. We also introduced our prototype system that implements five types of agents.

In the near future, instant messaging will be used as the communication façade for everything, such as coworkers, devices, things, services, stores, and so on.

References

1. Eulynn Shiu and Amanda Lenhart: How Americans use instant messaging, Pew Internet & American Life Project (September 2004), http://www.pewinternet.org/
2. Wikipedia, http://en.wikipedia.org/wiki/Instant_messaging
3. Isaacs, E., Walendowski, A., Ranganathan, D.: Mobile instant messaging through Hubbub. CACM 45(9), 68–72 (2002)
4. Isaacs, E., et al.: The character, functions, and styles of instant messaging in the workplace. In: ACM Conf. on Computer Supported Cooperative Work, pp. 11–20 (2002)
5. Nardi, B.A., Whittaker, S., Bradner, E.: Interaction and outeraction: instant messaging in action. In: ACM Conf. on Computer Supported Cooperative Work, pp. 79–88 (2000)

6. Rebecca, E.: Grinter and Leysia Palen: Instant messaging in teen life. In: ACM Conf. on Computer Supported Cooperative Work, pp. 21–30 (2002)
7. Huang, E.M., Russell, D.M., Sue, A.E.: IM here: Public instant messaging on large, shared displays for workgroup interactions. In: Proc. of the SIGCHI Conf. on Human Factors in Computing Systems, pp. 279–286 (2004)
8. Peddemors, A.J.H., Lankhorst, M.M., de Heer, J.: Presence, Location, and Instant Messaging in a Context-Aware Application Framework. In: Chen, M.-S., Chrysanthis, P.K., Sloman, M., Zaslavsky, A. (eds.) MDM 2003. LNCS, vol. 2574, pp. 325–330. Springer, Heidelberg (2003)
9. Rovers, A.F., van Essen, H.A.: HIM: a framework for haptic instant messaging. In: CHI 2004 Extended Abstracts on Human Factors in Computing Systems, pp. 1313–1316 (2004)
10. Roychowdhury, A., Moyer, S.: Instant Messaging and Presence for SIP Enabled Networked Appliances. In: Proc. of Internet Telephony Workshop, pp. 73–79 (2001)
11. Aurell, S.: Remote Controlling Devices Using Instant Messaging. In: Proc. of the 2005 ACM SIGPLAN workshop on Erlang, pp. 46–51 (2005)
12. Foley, C., de Leaster, E., van der Meer, S., Downes, B.: Instant Messaging as a Platform for the Realisation of a true Ubiquitous Computing Environment. In: Innovation and the Knowledge Economy Issues, Applications, Case Studies, Part 2, pp. 1051–1060. IOS Press (2005)
13. Adams: You Have Mail! (2001), http://www.openp2p.com/
14. Rodríguez, M., Favela, J.: Autonomous Agents to Support Interoperability and Physical Integration in Pervasive Environments. In: Menasalvas, E., Segovia, J., Szczepaniak, P.S. (eds.) AWIC 2003. LNCS (LNAI), vol. 2663, pp. 278–287. Springer, Heidelberg (2003)
15. Rodríguez, M., Favela, J.: A Framework for Supporting Autonomous Agents in Ubiquitous Computing Environments. In: System Support for Ubiquitous Computing Workshop at UbiComp 2003 (2003)
16. Rodríguez, M., Favela, J., Gonzalez, V.M., Muñoz, M.A.: Agent Based Mobile Collaboration and Information Access in a Healthcare Environment. e-Health: Application of Computing Science in Medicine and Health Care 5, 133–148 (2003)
17. Murph, D.: Samsung prepping RFID-enabled refrigerator (2007), http://www.engadget.com/2007/01/04/samsung-prepping-rfid-enabled-refrigerator/
18. Day, M., Aggarwal, S., Mohr, G., Vincent, J.: Instant Messaging / Presence Protocol Requirements, RFC 2779 (2000), http://www.ietf.org/rfc/rfc2779.txt
19. Milojicic, D.S., et al.: Peer-to-Peer Computing, HP Laboratories, Palo Alto, Technical Report HPL-2002-57 (2002)
20. Buschmann, F., et al.: Pattern-Oriented Software Architecture, vol. 1. John Wiley & Sons, Chichester (1996)
21. Extensible Messaging and Presence Protocol (XMPP) (2004), http://www xmpp. org/
22. Booch, G., Rumbaugh, J., Jacobson, I.: The Unified Modeling Language User Guide. Addison-Wesley, Reading (1998)
23. Hazas, M., Scott, J., Krumm, J.: Location-Aware Computing Comes of Age. Computer 37(2), 95–97 (2004)
24. Jabber, http://www.jabber.org/
25. Jsh: OpenSource Java Shell, http://gerard.collin3.free.fr/

Developing a Wearable Assistant for Hospital Ward Rounds: An Experience Report

Kurt Adamer[4], David Bannach[1], Tobias Klug[2], Paul Lukowicz[1], Marco Luca Sbodio[3], Mimi Tresman[5], Andreas Zinnen[2], and Thomas Ziegert[2]

[1] University of Passau, Germany
[2] SAP Research, Germany
[3] Hewlett Packard, Italy Innovation Center
[4] Steyr hospital, Austria
[5] Edna Pasher PhD. and Associates, Israel

Abstract. We describe the results of a three year effort to develop, deploy, and evaluate a wearable staff support system for hospital ward rounds. We begin by describing elaborate workplace studies and staff interviews and the resulting requirements. We then present a wearable system developed on the basis of those requirements. It consists of a belt worn PC (QBIC) for the doctor, wrist worn accelerometer for gesture recognition, a wrist worn RFID reader, a bedside display, and a PDA for the nurse. Results of evaluation of the system, including simulated (with dummy patient) ward rounds with 9 different doctors and accompanying nurses are given. The results of the evaluation have lead to a new system version aimed at deployment in real life 'production environment' (doctors and nurses performing ward rounds with real patients). The paper concludes by describing this next generation system and initial experiences from a first two week test deployment in a real life hospital setting.

1 Introduction

Hospitals are complex, yet highly structured environment where electronic information collection, processing, and delivery plays a central role in virtually all activities. While the introduction of electronic records has improved quality and enabled more efficient information exchange between institutions, it has in many ways actually complicated the work of the hospital personnel. The core of the problem lies in the fact that complex interaction with a computer system is not possible during most medical procedures. Clearly it is hard to imagine a doctor dealing with a notebook or a PDA during surgery. However even during simpler procedures access to computers is problematic. This is due to several factors. First of all, examinations are often performed on thigh schedules with doctors having just a couple of minutes per patient. Quoting one of the interviewed doctors they want: 'time for the patient, not time for the notebook'. Second, the patients expect doctors to devote their full attention to their problems. Retaining the 'human touch' and showing concern were major issues voiced by doctors

C. Floerkemeier et al. (Eds.): IOT 2008, LNCS 4952, pp. 289–307, 2008.
© Springer-Verlag Berlin Heidelberg 2008

interviewed in our work. Thus doctors are reluctant to interrupt the examination and focus on a computer instead of the patient, even if is only for a very short time. Finally examinations require hands to be sterile. This means that hand based I/O (mouse, keyboard, touchscreen etc.) is not feasible in between examination steps. As a consequence of the above issues electronic information access is severely restricted during dealing with patients. Patient data is often simply printed out before a procedure and taken along in paper form. Information is also rarely entered directly into the system. Instead it is entered into dictating machines, written on paper by accompanying nurses, or even stored as 'mental notes'.

This paper describes the results of a three year long effort to develop, deploy and evaluate wearable information access system for hospital staff. The work has concentrated on so called *ward rounds*, which are crucial to all hospital activity, and impose particularly stringent requirements on mobility, unobtrusiveness and cognitive load issues. The work described in this paper has been developed within the EU funded wearIT@work project [1].

1.1 Related Work

The only wearable solution for a ward round scenario has been investigated by the E-Nightingale project. It has looked at wearable monitoring of nursing activities to prevent accidents [2]. A number of usability studies were performed with respect to the use of different mobile devices. A PDA vs. notebook for accessing patient's records study was undertaken by [3]. It shows that physicians are generally more satisfied using the notebook computer although pointing-and-clicking applications are faster on PDAs. In [4], PDAs were compared to notebooks for nursing documentation application. The results also indicate that with the exception of writing notes, the overall user satisfaction is very similar for the both systems. Another evaluation of hand-helds for nursing was done by [5]. Many mobile and tablet based solutions were proposed and implemented. Examples include a prototype mobile clinical information system [6] and the WARD-IN-HAND project [7]. With respect to novel interfaces a glove-based gesture interface is evaluated in [8]. Finally a commercial product in form of a tablet PC optimized for hospital environments has been introduced by Intel (Intel mobile clinical assistant [9]).

1.2 Paper Contributions and Organization

The focus of the paper is not as much on the technological solutions, but on the experience made with design and evaluation of pervasive computing system for and in a complex real world environment. Section 2 describes an elaborate (over 1 year long) workplace study that included observation of ward rounds and interviews with all involved persons. The derived system requirements exemplify the benefits that hospital staff expects from pervasive technology and the boundary conditions for its deployment. We then described a system that we have implemented on the basis of this requirements. Detailed evaluation of the

system with doctors and nurses is then given. We conclude with a description of a next generation system based on this evaluation. We sketch initial experience with the deployment of this system in real life 'production' environment (ward rounds with real patients during normal hospital workday).

2 Application Design

The design and development of the wearable system for ward rounds followed an ISO 13407 compliant UCD (User Centered Design) process model. The different methodologies discussed in this section covered the first iteration of the process.

2.1 Field Observations

Early in the project a series of observations were conducted, where four doctors from two different departments were shadowed by two researchers during their daily ward round, two from the surgery department and two from the internal medicine department. The observations aimed at identifying typical activities during a ward round, and how these were distributed between doctor and nurse. During the observations, a number of challenges were encountered. First of all, the environment was completely unknown to the observer. When observing office users, the basic setting is already known in advance and the observer has first hand experiences with this kind of settings. As a consequence, it was difficult to assess the importance of events and environmental factors as well as note the right things. The speed at which unknown activities take place is another challenge the observer has to cope with. The ward round is a highly optimized and parallelized process. For example, a doctor might read the patient's file, order an examination, talk to the patient and document his visit, all in a matter of seconds. In a typical approach, one would interrupt the doctor and ask for clarification when things went too fast and were not understood. However this approach is prohibited by the environment that is studied. The doctor is not performing the ward round in his office, but in the hospital based on a strict schedule. In our scenario the hospital administration did not approve any technique that would have interrupted the ward round or would have made the patient aware of the fact that a study was in progress. For this reason and also for privacy reasons, audio and video recordings were not possible. Therefore the only option was to conduct retrospective interviews after the observations if the doctor's did not have other appointments which is typically the case after the ward round.

Jotting field notes on paper worked for the initial observations when trying to uncover the breadth of activities and environmental influences. However, during later observations when the goal was to capture the sequence and typical duration of events, manual note taking was not appropriate any more. During these observations, a Tablet PC running the custom TaskObserver software [10] was used to assist taking these kinds of notes. This solution allowed capturing of good data even though video recording was impossible.

In the following step the findings were used in order to develop first design ideas. These Mock-ups and the respective results of their evaluation are described in the following section.

2.2 Mock-Up Design and Evaluation

Based on observations and interviews, first design ideas were generated. Those ideas were evaluated using a mock-up system as described in [11]. The system allowed the doctor to view patient records on a Tablet PC and gave the nurse the opportunity to complete an x-ray request on a PDA. The studies, which were done using the mock-up system, pursued three goals:

1. Evaluate the feasibility of using the selected devices during the ward round.
2. Analyze whether the distribution of tasks between doctor and nurse was practical.
3. Compare speech and pen input as possible input modalities for the Tablet PC.

The evaluation was focused on following four ward round activities: (1) accessing patient data, (2) accessing individual findings, (3) organizing a laboratory analysis, and (4) organizing an examination.

The above mentioned activities form a representative subset of all the different tasks which have to be done during the ward round, assuming that the input and output modalities and the workflow of the other activities would be similar to those that have been tested.

The mock-ups were constructed using Macromedia Flash. The ward round activities obviously require collaborative work between doctors and nurses. In order to support collaboration, two mock ups were created. These were installed on a tablet PC and on a PDA for the doctor and nurse respectively. The tablet PC was remotely controlled (VNC) to simulate missing system functions such as: identification of the patient, automatic display of patient information as a response to identification and communication between the tablet PC and the PDA.

Medical data and findings of a fictitious patient were entered to assure a realistic feeling while interacting with the system. The realistic medical content of the system was created with the support of the medical staff of the Steyr hospital. The participants in the study were five doctors: two surgeons, one orthopedist and two internists. Two nurses were assisting, one for the surgeons and the orthopedist, and one for the doctors from the internal medicine department. The tests were performed in a patient room at the hospital, but without any patient being present. The participants were introduced to the use of the tablet PC and the PDA. Thereafter, the evaluation was carried out. Therefore, the doctor/nurse pairs were asked to role-play a typical ward round using the mock-ups available. Comments, opinions and suggestions from the participants were encouraged. Although discussions during the role-playing interrupted the workflow to a certain extent, it was important to allow immediate feedback to avoid incomplete response after the tests.

The mock-ups were complete enough to give a basis for discussion and answers to certain questions, but incomplete enough to allow room for comments and new ideas. The setup also included enough context to bring up relevant feedback about practical aspects of the workflow and devices. The Tablet-PC was regarded too heavy by most participants. Additionally, the device interfered with the physical patient-doctor interaction because there was no place to put it down in a patient room where it would not have been contaminated. These aspects would practically not have been discovered without doing tests within the considered context.

2.3 Functional Requirements

The field observations and the evaluation of the mock-ups resulted in a set of 57 requirements. In the course of the project, those requirements were iteratively adapted and specified in a more detailed way. As not all requirements could have been implemented within the scope of the wearIT@work project, the most important and manageable from an end user perspective ones were selected. These requirements are described in detail in the following:

Input Interface. Users must be able to switch to hands-free use for short periods of time. All time hands-free use is not necessary, but changing the interaction type has to be easy and fast. Doctors frequently perform examinations on patients which require them to use their hands. Once they have touched the patient, the hands are contaminated and have to be disinfected before using any device again. Finally, the interaction with the system must allow actions to be performed in a minimal amount of steps.

Content display. System must be able to display different types of content such as text (examination and lab reports), audio (recordings of examination questions and voice mails) or pictures (photos of wounds, ultrasonic examinations) - x-ray pictures (these require special treatment because of special requirements to resolution and size) - charts (vital data over time). It would also be desirable for the system to automatically adapt the information display to what the doctor is currently doing.

Information sharing. Doctor must be able to share and discuss information with colleagues and patients. The doctor often wants to show the patient for example x-ray pictures in order to help her understand her own condition better. The doctor often involves the junior doctors in diagnosing patients, and the information (the same display) should thus be visible to them all as a basis for discussion/ teaching.

Collaborative workflows. Users must be able to perform workflows cooperatively. Example: Several pieces of information are necessary to order an examination. The system should enable the doctor to start an order and fill in the desired information. The incomplete order is then automatically passed to an accompanying nurse to complete the request. Both doctor and nurse should be able to collaboratively edit and complete the order.

Authorization by proximity. The system needs to allow the following scenario: Person A and B are standing close together. A is authorized to do a

specific task, but B is not. The proximity of B to A should authorize B to do the task as well, because it is assumed that A issued a vocal command to B thus indirectly authorizing B. In the healthcare scenario this maps to the relationship between doctor and nurse who can jointly fill in for example examination orders.

Patient identification. The system must be able to automatically identify patients that are close (1-3m) to the wearable user. Patient identification is needed to avoid mix-ups. Doctors cannot rely on the patient being able to state his name. A beacon on the patient should take the form of a wristband. The patient should not have to remove the beacon at any time (e.g. during showers).

Personal patient contact. The system should not hinder personal contact with the patient. The input/output modalities of the system should enable the doctor to give most of his attention to the patient, and not to the system. Therefore, voice commands to a computer system are inappropriate while a patient is present.

Sound recordings. The system should be able to record a dictation of free-text messages and enable the recipient to play them. Voice messages should also be available for listening in mobile situations. Shorter messages can be recorded without form restrictions.

Ability to make photos. Some device is necessary that can take photos of a reasonable quality. The system needs to attach context information to these photos and store them automatically. Example: When a doctor examines a wound she often wants to document the current state. A photo taken at this moment should automatically be associated with the patient and date and attached to the patient file.

Unobtrusiveness. It should be possible to wear the devices so that contamination is avoided. The system should enable interaction during a physical examination, i.e. when hands are contaminated, without risk of spreading contamination to other parts of the doctors body.

The physical properties of the system components should make use and transport easy and comfortable. System components should be light (lighter than a tablet PC), easy to store during movement (e.g. in a lab coat pocket), not become warm during use (when handheld), not contain too many small loose parts (these can get lost).

3 First Demonstrator System

This section will briefly describe the first demonstrator system which was built on the basis of the above requirements.

3.1 Scenario Description

Fig. 1 shows an overview of the first prototype. The doctor and nurse teams move around without paper, laptop or filing card. The doctor identifies the patient by

means of his RFID reader on his wrist. The relevant patient files appear on the screen, attached to the bed, which at other times could be used as a television or an internet terminal. The doctor immediately gets an overview of all the important data about the patient, most recent results, current medication, x-rays or operation reports. With the inertial sensor attached to his wrist, he can navigate through the application without a mouse or keyboard. This is keeping his hands sterile and free to examine the patient at any time. Via his headset, the doctor can record new findings about x-ray requests related to his patients. The nurse, next to him, receives the information on her PDA, where she can view the diary of the x-ray department and directly arrange an appointment for the patient.

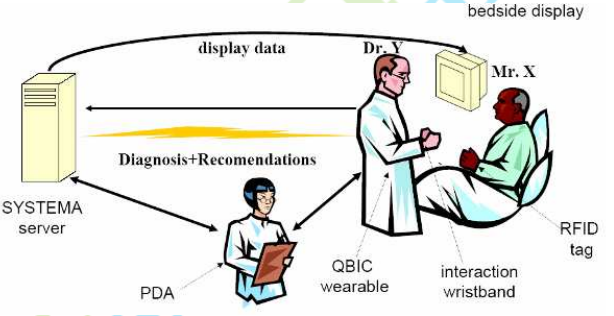


Fig. 1. The scenario behind our system

3.2 System Architecture

The first prototype consisted of three main subsystems; the doctor system, the bedside system and the nurse system (see Fig. 2).

Doctor Subsystem. The doctor subsystem consists of an unobtrusive wearable computer, an interaction wristband, and a headset for audio in- and output.

1. Wearable Computer. The wearable system is the doctors personal information gateway. It controls the I/O devices, all the sensors, it performs most context recognition tasks, and it provides network connectivity (with the nurse, the bedside system and all hospital databases). We chose the QBIC wearable computer, which features a 400 MHz Xscale processor, Bluetooth and USB interfaces, and a Debian Linux OS, all integrated in a wearable belt. A USB WiFi adapter is used for communication with the other subsystems.
2. Interaction Wristband. The interaction wristband is used for patient identification using RFID reader, and it provides a simple gesture interface as described in [12]. We used a small RFID-reader module attached to a USB-powered 'tmote sky' sensor node. For gesture recognition we used the MT9 motion tracker sensor from Xsens. It features accelerometers, gyroscopes, and magnetic field sensors, all on three axis and sampled with 100 Hz. The sensor was attached to a Bluetooth module including a battery pack.

3. Audio headset. For speech recording a wireless headset was used. Where a headset is not possible (because of the need for a stethoscope) a chest worn microphone will be used.

The doctor subsystem acts as the main gateway between sensors and the other subsystems. It gathers and processes the information from sensors, and sends suitable commands to the nurse- and bedside subsystem.

The gesture recognition and sensor processing software on the doctor subsystem was implemented using Context Recognition Network (CRN) Toolbox [13]. The CRN Toolbox basically consists of a set of parameterizable algorithms that can be applied on data streams. An application is defined by selecting the algorithms to be applied, defining the parameters of each algorithm, and connecting the input- and output streams to form the path of the data streams.

Nurse Subsystem. The nurse subsystem consists of an application server and a PDA as a thin-client frontend. It allows the nurse to enter and retrieve parts of the information relevant for her tasks, and to use those feature of the system, which are too attention-consuming and require too much manual interaction for the doctor to perform.

The nurse subsystem runs an application server which receives messages from the doctor subsystem and sends according forms to the PDA of the nurse which, in turn, can review the form, enter additional information and send it back to the application server. Also, if a dictation of the doctor is recorded, the application server can forward the audio file to the nurse's PDA. The nurse subsystem is based on MS Windows.

The Bedside Display. The bedside subsystem runs a Java application which is simulating the document browser of the hospital information system. The browser is able to display different types of content. Examples of the manageable content so far are text documents like examination and laboratory reports, audio recordings of examination questions on the PDA, pictures of wounds or x-ray pictures. An icon on the top right corner displays the current activation state of the gesture recognition. Our document browser can be controlled with four basic commands (scroll up, scroll down, open document, close document). Further commands set the enabled/disabled state of gesture control and the display of the browser itself.

Gesture Recognition. For being able to control the document browser by gestures our system on the doctor's wearable needs to recognize at least four different gestures. The gestures should be simple to perform on one side, and on the other side, they must be unique enough to be distinguishable from all other human movements. For easing the second requirement we introduced two special gestures for activating and deactivating the recognition of the other, simpler to conduct gestures.

Meeting these requirements we defined the following six gestures that can be conducted by swiveling the forearm up and down, left and right, or by rolling it left and right respectively:

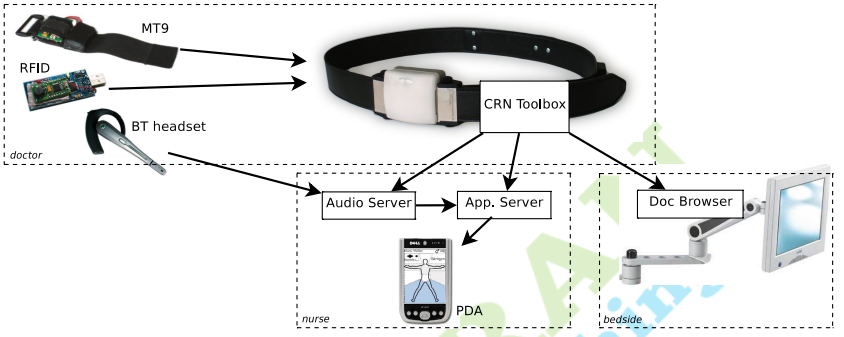


Fig. 2. System architecture of the first demonstrator

Forearm Gesture	*Command*
swivel up, then down	UP
swivel down, then up	DOWN
swivel left, then right	OPEN
swivel right, then left	CLOSE
roll right, then left, twice	ACTIVATE
roll left, then right, twice	DEACTIVATE

We use the three-axis gyroscope of the MT9 motion sensor to detect the swiveling and rolling of the forearm. The x axis of the gyroscope is aligned parallel to the main axis of the forearm, and the y axis is parallel to the plane of the hand. Rolling the forearm leads to either a positive or negative deviation of the angular velocity on the x axis, depending on the roll direction. Similarly, swivel left/right is measured on the y axis and swivel up/down on the z axis. The events of turning in either direction are extracted from the raw data stream by a threshold filter. The sequence of these events is analyzed and compared to pre-recorded sequences of valid gestures. Detected gestures with overlapping sequences are discarded.

4 Evaluation of the First Demonstrator System

The following section summarizes the results of a first system evaluation. The first subsection describes initial experiments with students. Afterwards, the system evaluation in a hospital in Steyr is described including the evaluation environment, training and test setup. Applied evaluation methodologies are introduced. The section ends with the main findings during the evaluation.

4.1 Student Experiments

Doctors are usually rarely available for in-depth user studies. From our experience this user group becomes very soon frustrated by a new system which is

not functioning as expected. Therefore, we decided to first conduct user studies with students. The goal of these tests was to identify as many potential usability problems as possible, so that we were able to refine the prototype for the usability studies with our real end users.

Our test setup was designed to resemble the setting of the clinical ward round as closely as possible with respect to the interaction between user and system. The main activity that requires user interaction with the system during the ward round is a doctor browsing the patient health record. We have mapped the interaction necessary to navigate the patient health record into a domain where students that do not have a degree in medicine feel more comfortable with. Our test application lists headlines of news stories that can be opened to read the full text including related pictures. The interaction remains the same, but the news documents are meaningful to the test users, whereas medical records could have confused them. The computer display was positioned on a high desk, requiring the user to stand in front of the system.

The study required the user to switch frequently between system interaction and a real world task. The evaluation of the studies contained two parts. One was an analysis of video recordings towards obvious usability issues. The second part was an analysis of interviews we conducted with each person after performing the test. People were asked to explain what they did and did not like about the system.

In total 23 students from the local technical university took part in our tests. The analysis of these studies revealed a number of minor and some major usability issues. One of our findings was that the test participants felt that the *up* and *down* commands were easier to perform and more intuitive than the other gestures. This may relate to the fact that the up and down gestures are more naturally mapped to the visual feedback that the application gives (i.e. moving the hand upwards or downwards results in an upward or downward movement of the highlighting of the respective document on the screen). The gestures for opening and closing a document (i.e. moving the hand to the left and to the right), however, as well as activation and deactivation (i.e. turning the wrist twice to the right or twice to the left), are not naturally but rather semantically mapped to the application. In other words: these gestures were mapped to a specific function in the application and not to the same movement. These gestures were mainly chosen because of their simplicity, but partly also with the metaphor open and close book for open and close, and turn a simple electrical device on and off (using a knob) for activate and deactivate. These metaphors were not clear to all participants; therefore most participants found these gestures less intuitive than the more naturally mapped open and close. Another metaphor for open and close mentioned by one participant was the use of a door.

One of the most important insights from the results of the experiments was that it was very difficult for the test participants to understand exactly how they were to perform the gestures in order for the system to understand these correctly. One problem was that the time available for learning in the experiment was too short for the participant to have a chance to learn the gestures well

enough to be able to perform the given tasks reasonably efficiently. The other problem was that our team, acting as test coordinators, had trouble communicating our knowledge on how to perform gestures to the test participants, even if we knew how to perform them effectively. The reason for this is that a large part of this knowledge is implicit, i.e. cannot really be described verbally since it is more or less tacit knowledge. The most effective way to gain tacit knowledge is to practice. For the following evaluation with doctors, we increased the training time. Furthermore, we applied the gained experiences in explaining and giving them metaphors to remember the gestures.

4.2 Setup of Experiments with Doctors

For evaluation, a patient room was prepared to test the prototype. The fixed installation included the bedside display attached to the patient's bed, and a patient doll from the medical training department, that was equipped with a RFID wristband for identification. The doll was used as a substitute for a real patient to simulate physical interaction with the patient. The technical infrastructure necessary to keep everything running was positioned on a table in the corner of the room. A video camera constantly captured the doctor during his interactions with the bedside display and the patient.

End users tested the system in doctor-nurse pairs. A total of 9 doctors from several departments took part in the study. The doctor was wearing the interaction wristband and a Bluetooth headset for audio in- and output. The nurse was given a PDA. Each evaluation session started with an introduction to the project. After a short explanation of the prototype, the doctors and nurses trained to operate with the system. The training session was split into two halves. During the first part, the doctor and nurse were introduced into the systems usage and asked to familiarize themselves with the interaction methods. After a certain degree of proficiency with the system was reached, a sequence of activities similar to that of the actual tests was explained and performed. During the training, instructions were given to the participants. Subsequently, an evaluation sequence was performed three times. After the training session, the doctors and nurses entered the room through the door. Next to the patient's bed, the doctor had to use the interaction wristband to scan the patient's identification wristband. The system brought up the patient's files on the bedside display. The doctors had to open a document and scroll to its end containing instructions about the next steps, e.g. opening another document. The last document contained instructions to perform an examination of a certain patient's body part. In a last step, the doctor issued an examination request that was completed by the nurse. Finally, after the practical part of the experiment, the participants filled out a questionnaire concerning their experience using the system and an interview was conducted.

4.3 Evaluation Method

The purpose of the test was to evaluate how well the nine participating doctors and nurses could interact with the system in the context of a typical ward round work flow.

In order to analyze the performance of the participants, identify problems, and capture spontaneous comments, each test sequence was entirely filmed. At a later point in time, practical interaction problems, conflicts between system interaction and work flow and comments were analyzed and documented.

Additionally, the participants filled out a questionnaire concerning their experience using the system. The questionnaires were constructed using Likert scales meaning that the test participants were shown a list of statements like *I found the gestures easy to perform* or *using this system could make my work more efficient*. The participants had to categorize on a scale between 1 and 5 to what extent this was true. The participants were also interviewed in order to find out more details about their opinions and ideas concerning the system and its use.

4.4 Results

In the following, the main results of the evaluation are summarized. In a first paragraph, overall system results are presented from doctors' and nurses' point of view. A second paragraph steps into the findings using gestures as an interaction method.

Overall System Results. When asked if they would use a system such as represented by the prototype if it was available, the doctors were hesitant, and gave on average neutral answers (3+/5 on Likert scale). The doctors were not fully convinced that such a system would make the ward round easier and more efficient. Most likely, the skepticism can be explained by the novelty of the gesture interaction, which is a central part of the prototype. Several doctors were reluctant to the idea of performing gestures in front of the patient, concerned that the patients would consider them crazy or strange, waving around with their hands in the air. According to the doctors, attention for interacting with the proposed system would decrease the patient contact. The doctors were only mildly thrilled (3+/5 on Likert scale) to perform extensive system training. However, several doctors could imagine using gesture interaction if the gestures were smaller and/or more elegant. Furthermore, some doctors commented that hands-free interaction was good because of hygiene reasons, and that the negative implications of the gesture interaction would most probably decrease with more practice.

The possibility to view the latest patient documents electronically, and to be able to enter fully authorized examination request immediately into the system, was seen as very positive and helpful for the efficiency of work, because it would reduce unnecessary paper work (post ward round) and duplication of information. The doctors on average (4/5 on Likert scale) found the audio feedback (start/end/error dictate audio recording) very helpful. Almost all doctors agreed that speech input was a suitable method to perform the examination request (in average 4+/5 on Likert scale). This probably has its cause in the fact that the doctors currently dictate the examination request to the nurses who write down the information. Basically, there is no great principal difference dictating to the system instead of the nurse.

Generally, the nurses were very positive. It should be said, however, that the nurses had quite small role in the experiment and their tasks were not very extensive, although still representative. Further, the time pressure that is often present during a real ward round was not part of the experiment setting. When asked if they would use the system if it was available, the nurses answered with a clear yes (4/5 on Likert scale). They felt that their ward round related work would go faster and be easier with the use of such system (4+/5 on Likert scale) connected to the hospital's backend. Today, in fact, the nurses process most of the information collected during the ward round only after it is finished; such situation could improve with the introduction of a fully functional version of the tested system. The nurses felt to have the system under control (4 /5 on Likert scale). The comments revealed that they found the interface easy to understand, and the touch screen interaction simple to perform. According to the nurses, they would have no problem controlling the interface perfectly (meaning a 5/5 answer to the control question) with a little more training.

The contact with the patient is to all nurses extremely important. Thus, several of them are worried that the new technology would take position in between themselves and the patient, and that they would be too busy with the PDA, especially at the beginning before the interaction becomes more automatic, but to a certain extent also after the initial stage. Especially the nurses from the pediatric department thought that the system would be a change to the worse and even unacceptable. Today, the only responsibility of the pediatric nurses is to talk to the children and comfort them during the ward round. Any further tasks that would take attention away from the children are not welcome. On the other hand, some of the nurses (non-pediatrics) felt that there was no greater difference using the PDA compared to taking notes like today, and that both tasks would require about as much attention. In average, the nurses answered slightly negatively to the question if the new system could help improve the patient contact (0-/5 on Likert scale).

Gesture Interaction Results. The gesture interaction was by far the most controversial aspect of the prototype, which is not surprising considering the novelty of the concept, and that none of the test participants had had experience with anything similar before. Below, the most frequently occurring issues and their potential causes are discussed. Most of the problems will most probably become less frequent when the test participants were given the possibility to practice the gestures for a longer period of time. Due to the short training-period included in the experiments, most of the participants did not reach a level of proficiency where they could work comfortably. However, a certain learning effect can already be observed from the tests results.

Shape: 6/9 doctors had problems performing a gesture without striking out in the other direction first in order to get power for the movement. This often led to a situation where the opposite gesture was recognized, i.e. the cursor moved up instead of down or a document that was meant to be opened remained closed (since the command close will have no effect on an already closed document). This issue was mainly caused by the fact, that doctors had

no direct feedback, i.e. the effect of striking out is not perceived while doing it, only the end-result is seen. 4/9 doctors sometimes forgot to perform the second part of the gestures, i.e. the movement back to the original position. One reason for this mistake is that doctors found the back movement less intuitive. This is because the movement goes in the direction opposite of the desired effect. This is the case at least for the up-down gestures, i.e. in order to move the cursor up, one must move the hand up but then also back down.

Timing: 6/9 of the doctors had problems performing the gestures in the right tempo. Most frequently, they carried them out too slow, i.e. under the threshold speed defined in the gesture recognizer. However, a clear learning effect was observed from the beginning of the experiment to the end. In the initial practice phase, 5 doctors performed the gestures too slow, while during the third and last experiment task, only one doctor made this mistake. Some doctors also performed the gestures too fast to be recognized (3/9).

Conceptual: Most of the doctors at some point confused the gestures. 2/9 doctors at some point forgot completely how to perform a gesture, or at least had to think for a long time before remembering it. 7/9 confused the paired gestures with one another (for example open with close or activate with deactivate). After a bit of practice, however, the confusion decreased, and in the last phase only 1 doctor confused the gestures with each other. The up and down gestures were rarely confused at all.

A very frequent problem was that doctors completely forgot the gestures while activating/deactivating the system (i.e. doctors tried to interact with the system without activating it, or started with non-system-related activities, like examining the patient, without deactivating). A possible reason for this is the fact that doctors did not really see the benefit of being able to move their hands freely without their movements being randomly recognized. Probably, this was because the experiment context was not realistic enough in terms of activities involving the hands. For example the test did not contain a full patient examination (most doctors only symbolically touched the fictitious patient when asked to do so) or lively communication with either nurses, or colleagues, or the patient. Consequently, doctors did not really see a need for the de-/activation action and thus forgot about it. All 9 doctors at some point in time had this issue.

In summary, the use of arm gestures was the most controversial part of the tested system. As doctors had never used such an interaction technology before many beginners mistakes occurred, like performing gestures to fast or to slow, as well as confusing gestures or just forgetting how to perform a particular gesture.

Three issues regarding the gestures were found to be important for the further development of the prototype:

- Gestures have to be socially acceptable: There were several comments that doctors do not want to perform gestures in front of patients because of the fact that these might be distracting or doctors just felt stupid performing such gestures.
- Gestures should be easy and simple to perform in a way that those gestures require not to much of the doctors attention. This issue became apparent

when doctors were asked whether gesture interaction was comfortable and making their job easier.
– There should be some sort of feedback, whether a gesture was performed right or not: As there was no direct visual feedback showing what the systems interpretation of the performed gesture was (i.e. when performing a gesture in the wrong way) doctors felt not to be in control of the system.

5 Next Generation Prototype

Based on the above finding a next generation prototype has been designed and implemented. A major aim of the implementation has been to facilitate deployment in real life 'production' ward rounds with real patients, as opposed to simulated ward rounds in the evaluation described in the previous section. Such deployment is currently under way and an initial two week test has just been completed. This section outlines the main features of the new prototype and discusses the lessons learned from the initial two week test.

5.1 System Overview

Compared to the first generation prototype the system has been modified in the following way:

1. It has been integrated with the existing hospital information system and its back-end technology. This has involved a major software restructuring to make it compatible with the requirements (see software description below). It has also lead to a modified network structure and application distribution.
2. The hardware has been modified for increased robustness (see hardware description below)
3. The gesture recognition has been improved according to the findings of the previous prototype evaluation. The improvements include better senitivity to allow tiny inconspicuous gestures, improved acceptance of variations, and optimized recognition speed for more immediate feedback.
4. A proximity detection system has been integrated to provide the contextual information about which persons (doctors, nurses) are taking part at each ward round.
5. Pictures taken by the nurse (e.g. of wounds) with her PDA are automatically attached to the patient's files in the hospital information system (just like the sound notes in the old system).

Hardware Components. The main hardware modification can be summarized as follows.

– Redesign and re-implementation of the wrist worn sensor modules. This includes (1) a new RFID reader with improved range and higher reliability, which ensure an easier and more accurate retrieval of patients identifiers, (2) a re-design and re-implementation of the Bluetooth based communication module to include an improved battery, charging circuits, battery and

function indicator and an external switch, and (3) re-packaging of the whole wrist band module in a more compact, easier to wear, and more nicely-looking component.
- The main processing unit worn by the doctor has changed from the QBIC system to a NOKIA 770, which, for our application, is only marginally more intrusive while having the advantage of being a 'mainstream', commercial device.
- An RF-based proximity detection system has been implemented and integrated, based on TecO Particles [14] which were attached to the badges of the hospital personnel.

Software Architecture. The overall software architecture of the prototype is shown in Figure 3. We can identify two main end-user applications the *MPA*

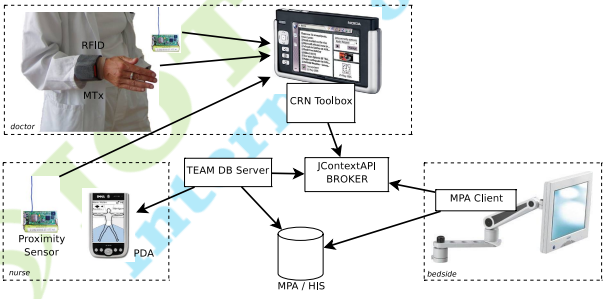


Fig. 3. The healthcare prototype system architecture

client application, which is used by the doctor to access the *Hospital Information System (HIS)* and the custom *PDA application*, which is used by the nurse to support her work during the ward round.

The MPA application (which is part of the hospital information system in Steyr) is provided by Systema (a partner of the wearIT@work consortium). It is a client-server application, which does not only allow the doctor to access all patients data, but also to order examinations, and to manage a number of other tasks related to daily activities. In our scenario, the front-end client of the MPA application has also been extended with two features: an interface to the gesture recognition system and an an embedded *Multimedia Server* (not shown in Fig. 3), which allows recording of the audio messages spoken by the doctor, and of the photos taken by the nurse.

To facilitate the integration with the existing HIS and MPA systems the distribution of context information is handled by *JContextAPI*, which is the Java implementation of the *Open Wearable Computing Framework (OWCF)* developed within the wearIT@work project. The core component of the *JContextAPI* acts as a broker to distribute context information to every registered consumer. In our prototype the JContextAPI broker sends proximity events, and patient identification events coming from the CRN Toolbox to both the MPA client and

the PDA application. It also notifies the MPA client of gesture events, so that MPA can react to doctor's gestures in a context-sensitive way. Furthermore, it propagates events produced by the Multimedia Server to notify when and where a new audio recording is available.

Compared with the previous generation system, the PDA application used by the nurse has been modified to handle photos and to interact with the OWCF.

5.2 Test Deployment

The real life deployment experiments are currently in progress. An initial two week test was performed in August 2007. The aims of the test were to identify any remaining technical and/or organizational issue specific to the real world environment. The deployment has taken place in a two-bed room (see Fig. 4) that over the time period was occupied by a total of 4 patients. During this time the round has been performed 20 times by two different doctors and 4 different nurses.

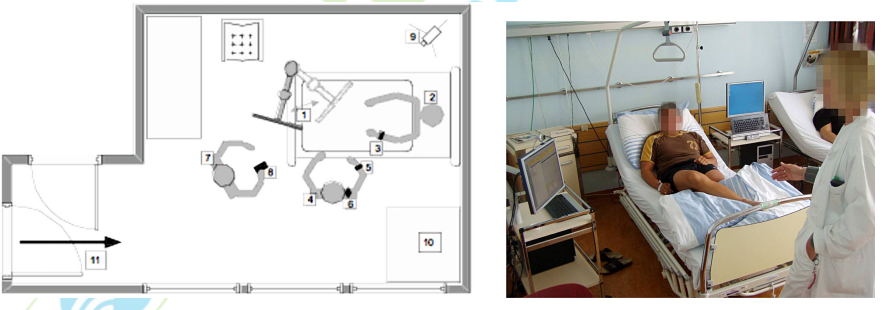


Fig. 4. Left: the evaluation setup for the first prototype. Right: The second generation prototype system in use in a two-bed patient room.

Deployment Issues. A number of issues emerged during planning for the initial deployment in the hospital. While they may seem trivial we choose to discuss them here as they are typical for moving from lab tests to real life deployment.

- Patient Considerations. To motivate the patients to participate in the experiments it has been suggested to allow internet access on the bedside systems. In addition our lab PCs had to be replaced by silent PC not to disturb the patients.
- Staff consideration. The staff had to be provided training on the use of the system including initialization, overall workflow and the use of gestures.
- Maintainability Considerations. During the deployment the system maintained by the hospital IT department. This meant that all system initialization procedures had to be properly automated and simplified. Protocols had to be established for daily maintenance including battery re-charging, rebooting systems and replacing components making problems. As will be described below the complexity and fragility of system setup components has been identified as a major source of problems.

Technical Lessons from Deployment Test. Despite intensive lab validation before the deployment, technical problems have invariably emerged during the test. Detailed analysis is currently underway. Key preliminary findings are:

1. The person independence of the gesture recognition was not satisfactory. While the gesture recognition worked fine for one of the doctors (as well as for test persons in our lab), the other had major problems handling it.
2. The procedures needed for system initialization and user registration were not robust enough. Obviously in many cases switching devices on/off, registering users, and coming into/out of range of Bluetooth nodes has been performed in sequence and at speeds which we did not consider in our lab trials. This often caused components to freeze.
3. Similar problems were caused by issues with wireless links. In particular when there were many people in the room (to view a demonstration of the system) Bluetooth links or the proximity detection signals would be blocked. In addition the way doctors were handling the wrist worn RFID reader combined with the new mounting often resulted in occlusions of the Bluetooth communication.

Initial User Feedback. Despite the focus on technical issues initial user evaluation was also undertaken. During this visit observations were conducted as well as interviews with two doctors, one of whom participated in the tests, and one who did not and spoke generally about working conditions and potential changes due to wearable computers. Additional interviews were conducted with two patients, who used the system during their stay, and with the hospitals technical specialist, who worked on servicing the system before and during the tests. The main results can be summarized as follows:

1. The staff reactions were strongly overshadowed by the technical issues described above. Clearly the problems with gesture training were, like in the first prototype, a key issue, and people suggested that speech control might be better. The fact that the system allows hands to remain sterile was nonetheless seen as positive.
2. The patients were positive about the system. They stated that seeing the results on the screen and having them explained in real time by the doctor increases patient's trust in staff. None of the patients interviewed felt such a system would have an effect on their relationship with staff, as the system will not prevent them from talking to the doctors and nurses. The patients interviewed stated that they would be happy to see such a system working in the hospitals in the future.

Acknowledgments

The authors would like to thank all those who have been involved in the work described in this paper. This includes in particular the wearIt@work partner companies TEAM and Systema, Steyr Hospital Team (Doctors, Nurses and IT department) and Peter Wanke from HealthInformatics R&D.

References

1. wearIT@work Project: http://www.wearitatwork.com/
2. Noma, H., Ohmura, A., Kuwahara, N., Kogure, K.: Wearable sensors for auto-event-recording on medical nursing - user study of ergonomic design. In: McIlraith, S.A., Plexousakis, D., van Harmelen, F. (eds.) ISWC 2004. LNCS, vol. 3298, pp. 8–15. Springer, Heidelberg (2004)
3. Rodriguez, N.J., Borges, J.A., Soler, Y., Murillo, V., Sands, D.Z.: A usability study of physicians' interaction with PDA and laptop applications to access an electronic patient record system. In: Computer-Based Medical Systems, 2004. CBMS 2004. Proceedings. 17th IEEE Symposium, pp. 153–160 (June 2004)
4. Rodriguez, N.J., Borges, J.A., Soler, Y., Murillo, V., Colon-Rivera, C.R., Sands, D.Z., Bourie, T.: PDA vs. laptop: A comparison of two versions of a nursing documentation application. In: Computer-Based Medical Systems, 2003 Proceedings. 16th IEEE Symposium, pp. 201–206 (June 2003)
5. Young, P., Leung, R., Ho, L., McGhee, S.: An evaluation of the use of hand-held computers for bedside nursing care. International Journal of Medical Informatics 62(2-3), 189–193 (2001)
6. Choi, J., Yoo, S., Park, H., Chun, J.: Mobilemed: A PDA-based mobile clinical information system. IEEE Transactions on Information Technology in Biomedicine 10(3), 627–635 (2006)
7. Ancona, M., Dodero, G., Gianuzzi, V., Minuto, F., Guida, M.: Mobile computing in a hospital: the WARD-IN-HAND project. In: ACM Symposium on Applied Computing (2000)
8. Tani, B.S., Maia, R.S., Wangenheim, A.v.: A gesture interface for radiological workstations. In: Computer-Based Medical Systems, 2007. CBMS 2007. Twentieth IEEE International Symposium, pp. 27–32 (June 2007)
9. Mobile Clinical Assistant, http://www.intel.com/healthcare/ps/mca/index.htm
10. Klug, T., Mühlhäuser, M.: Taskobserver: A tool for computer aided observations in complex mobile situations. In: Proceedings of the International Conference on Mobile Technology, Applications and Systems (Mobility), September 2007 (to appear)
11. Carlsson, V., Klug, T., Ziegert, T., Zinnen, A.: Wearable computers in clinical ward rounds. In: Proceedings of the third International Forum on Applied Wearable Computing, IFAWC 2006, pp. 45–53 (March 2006)
12. Carlsson, V., Klug, T., Zinnen, A., Ziegert, T., Levin-Sagi, M., Pasher, E.: A Comprehensive Human Factors Analysis of Wearable Computers Supporting a Hospital Ward Round. In: Proceedings of 4th International Forum on Applied Wearable Computing 2007, VDE (March 2007)
13. Bannach, D., Kunze, K., Lukowicz, P.: Distributed modular toolbox for multi-modal context recognition. In: Grass, W., Sick, B., Waldschmidt, K. (eds.) ARCS 2006. LNCS, vol. 3894, pp. 99–113. Springer, Heidelberg (2006)
14. TecO Particles, http://particle.teco.edu/

Social Devices: Autonomous Artifacts That Communicate on the Internet

Juan Ignacio Vazquez and Diego Lopez-de-Ipina

Faculty of Engineering
University of Deusto
Avda. de las Universidades, 24. 48007 Bilbao Spain
{ivazquez,dipina}@eside.deusto.es

Abstract. The Internet has boosted people collaboration, enabling new forms of exchanging knowledge and engaging in social activities. The Web 2.0 paradigm (also called the Social Web) has greatly contributed to this achievement. We believe that the next wave of smart devices and digital objects will leverage the pervasiveness of Internet connectivity in order to form ecosystems and societies of artifacts that implement Internet-based social behaviors and interact with existing Internet-based social networks. In this paper, we introduce the concept of *social device* and describe our experiences creating different kinds of augmented objects that use the Internet in order to promote socialization, look smarter and better serve users. We finally identify several challenges in the research of devices that behave in social ways.

1 Introduction

People are not isolated at their workplace or in public spaces: they meet, talk, exchange ideas and collaborate. In our modern world where no single person is likely to know everything about a particular domain, sharing information and cooperating is of foremost importance for the advance of the society.

In fact, complex tasks often require the combined contribution of different kinds of specialized knowledge and are also often more easily achieved as a collaborative activity. Nowadays, the concept of collaboration is itself undergoing a transformation driven by the new networked technologies. For example:

- Open Source communities have developed some of the most successful computer software applications in the last decade including GNU/Linux, Firefox, OpenOffice, Apache, Eclipse, and, more recently, the Java programming language that has also joined this trend.
- Collaborative content sites, created by the contributions of thousands of users, have been able to become the most important repositories of information in a surprisingly short time. The most popular examples are Wikipedia and YouTube.
- In P2P file exchange networks file sharing is promoted following the scheme "the more you share, the further ahead you are placed in download queues".

C. Floerkemeier et al. (Eds.): IOT 2008, LNCS 4952, pp. 308–324, 2008.
© Springer-Verlag Berlin Heidelberg 2008

The Web has transformed from a place with many readers and few publishers, into a space where everyone can have a voice via tools such as blogs (weblogs), or a community of users can collaboratively create a body of knowledge about some concrete topic via wikis.

This evolution of the Web has been dubbed Web 2.0 or Social Web, and it promises to boost human collaboration capabilities on a worldwide scale, enabling individuals and organizations to face challenges that could not have been faced in isolation. In the very same way, there is clear evidence that digital and electronic devices should not be isolated in the places where they are deployed or spontaneously meet.

We are witnessing an unprecedented increasing number of digital objects that are populating and almost invading more and more environments in an imperceptible way. Most of times, they carry out simple but efficient tasks that help people in their everyday activities, simplifying their lives. However, when it comes to complex tasks involving a higher-level of intelligence and cooperation in unexpected situations or scenarios, they invariably fail, because *they have not been designed for collaboration.*

Just as Web 2.0 technologies have boosted human social capabilities on the Web, devices can also benefit from similar schemes, using the Internet in order to communicate, collaborate, use global knowledge to solve local problems and perform in exciting new ways.

In this paper, we present some concepts about devices powered with Internet-friendly social capabilities, and describe our experiences. In section 2, we refer to previous work on integrating Internet, especially Web-based technologies, into everyday objects. Section 3 is devoted to the analysis of desired features of social devices that communicate through the Internet. Section 4 includes a description of several prototypes of devices we have implemented that use Internet to communicate with other entities. Finally, in section 5 we identify different challenges associated with the design of everyday objects that behave in social ways, which must be addressed by the research community.

2 Related Work

Internet and, particularly, Web technologies have been used in the past to create exciting Ubiquitous Computing scenarios where objects are gifted with augmented capabilities. Cooltown [9] [10] [1] was a pioneer project applying Web technologies to support users of wireless, handheld devices interacting with their environment, anywhere they may be. WSAMI [5] enables the creation of Ambient Intelligence scenarios controlled by Web-based invocation mechanisms, Web Services.

Remarkably, the last few years have witnessed an increasing interest in the application of a particular type of Web technologies, Semantic Web technologies, to create connected societies of smarter artifacts. Semantic Web technologies (RDF [23] and OWL [22]) have been used in several projects in order to provide more intelligence in environments. In the Semantic Web, URIs are used for

representing concepts, while HTTP [3] provides a natural means for retrieving RDF-based descriptions. Task Computing [13] [17] aimed at creating Semantic Web Services-based systems in which users could perform automatic composition of services, based on semantic service descriptions.

Other architectures such as CoBrA [2] provided both an architecture and an ontology (SOUPA) to create environments populated by smart devices that could communicate through the Internet using Semantic Web technologies. However, CoBrA requires a central server to be deployed where all the intelligence resides, acting the devices as simple slave entities out of the context-awareness process.

SoaM (Smart Objects Awareness and Adaptation Model) [18] [21] eliminated the need for a central component, and introduced the concept of collaborating *semantic devices*. The results demonstrated that spontaneous emergent intelligence could be achieved by the collaboration of individual objects that may access the Internet for retrieving data and ontologies.

Currently, one of the most popular approaches to implement Internet of Things experiences is the use of "touch computing" along with object tagging [16] [12] in order to obtain further information about a concrete object from the Internet. Mobile devices are generally used in these cases as service "activators".

However, the vision of a coherent and heterogeneous society of Internet-connected objects ("*from anytime, any place connectivity for anyone, we will now have connectivity for anything*" [4]) is in its first steps of realization. We think that current research must explore the possibility of embedding *social capabilities in objects and devices* in order to realize the "Internet of Things" vision as a replica to the current "Internet of People" reality. A lot of people use the Internet to find others' solutions to a particular personal issue; that is, they use global information for a local purpose.

In our vision, "social devices" should work very much in the same way. When facing a local problem, devices can "talk" with other artifacts that can provide their experience about that situation, or additional information that may help to come up with a solution. In the next section, we will describe the features that this kind of social devices must embody.

3 Features of Social Devices

Lassila and Adler [11] introduced the concept of *semantic gadget* to describe devices capable of performing "*discovery and utilization of services without human guidance or intervention, thus enabling formation of device coalitions*". Vazquez et al. [18] [20] introduced the concept of semantic device as a system "*able to spontaneously discover, exchange and share context information with other fellow semantic devices as well as augment this context information via reasoning in order to better understand the situation and perform the appropriate reactive response*".

The concept of *social device* emphasizes even more the benefits of the communication with other devices in order to determine the appropriate behavior. Internet is the enabling technology in this case, since the knowledge provided by local devices can be complemented with information obtained from the Internet (provided in turn by other devices, services, or humans). Further below, we will illustrate this situation with the example of a prototyped umbrella which is able to obtain current weather conditions through communication with surrounding rain sensors, but it is also able to download the weather forecast from the Internet (`weather.com`) in order to provide better advice to the user.

We have identified several characteristics that may help to define the concept of social device:

- Social devices are natively designed for collaboration
- Social devices integrate local and global information (obtained from the Internet) in order to apply community knowledge to particular problems
- Social devices are able to interpret all the exchanged information at a semantic level, whatever the vocabulary used

3.1 Designed for Collaboration

Social devices are inherently talkative: they are natively collaborative in the sense that they share all the information they can.

As already mentioned, the Web has transitioned from a basically "one publisher – many readers" model to a more collaborative "many publishers – many readers" model, in an approach that was called Web 2.0 [15]. The major representatives of this culture are weblogs, social bookmarking, wikis and RSS feeds.

We consider that this model can be also applied to devices, featuring a collaborative nature, sharing information, and creating a community of intelligent objects in the environment in order to better serve their users.

Higher and more useful knowledge can be obtained from the generous contributions of individual entities, rather than from selfishly not sharing information (of course, taking into account existing privacy concerns).

On the other hand, social devices can also interact with existing social networks (e.g., Facebook, YouTube, Flickr, Digg) in order to monitor some information on behalf of their user/owner. For example, a Flickr-enabled digital photo frame can periodically check whether new photos from the user's buddies have been published, download them and start a slideshow.

In this case, objects use existing social network infrastructure as a medium for retrieving user-related content. These social objects may also send data back to the social network (e.g., the user may vote for a photo in order to increase its popularity), thus becoming active participants in the community.

These scenarios illustrate how for some simple, well-defined activities users may interact with the social network via real objects instead of using a computer and a browser. These objects promote users' socialization in a transparent, unobtrusive way.

3.2 Integration of Local and Global Information

Users do not have the ability to connect their minds to the Internet, but devices can exhibit this feature. By managing more information they look more intelligent, as in the case of the weather-aware umbrella.

There are ubiquitous examples of this feature: a car may recommend the best route based on data provided by other surrounding cars and traffic information downloaded from the Internet; an eBay-aware ornament at the home entrance may project a green light beam if our bid was successful or a red one if not, as the user arrives home.

We have developed several prototypes of objects (described below in section 4) that simultaneously obtain context information provided by other local objects in the environment as well as remote services on the Internet, integrating all this knowledge to realize a higher level of context-awareness. In our prototypes local information was typically provided by wireless sensor network nodes, while global information was provided by public XML-based Web Services on the Internet. In both cases, we designed adapters that semantically annotated the data using RDF, and shared this information with all the entities in the environment, creating a distributed knowledge space, so existing objects could analyze the semantic context information and react in the desired ways [18].

3.3 Interpretation of Information at a Semantic Level

The vision of a myriad of social devices "talking" both at local and global level exposes the problem of the number of formats and vocabularies for exchanged information and messages. In our experiments [18] we used Semantic Web technologies, RDF and OWL, in order to support multiple vocabularies and dynamically learn relationships among concepts using ontologies.

We humans interpret the information on the Web at a semantic level, being able to understand, process and use it. Social devices should be able to perform in the same way: this is the main reason to apply semantic technologies.

Of course, this level of intelligence requires additional computation capabilities on devices (e.g., able to run a semantic reasoner), but we believe that semantic interpretation is a feature inherent to social communication, without imposing limitations in expressiveness or vocabularies.

The use of microformats [7] as an alternative provides the additional advantage that it is easier for humans to create microformats-enriched content than RDF-annotated content. This approach enables devices to interpret, at a basic level, user-generated content in blogs, wikis, and so forth, without demanding much effort from the contributors, thus bridging the communication gap between users and devices on the Internet.

4 Experiences Prototyping Social Devices

During the last years we have been researching the implications of Internet connected objects and their interaction with users. We have also developed several

prototypes in order to evaluate their potential. The following subsections describe some of these prototypes.

4.1 Flexeo's SensorMap: A Wireless Sensor Network –Based Mash-Up

The goal of the project Flexeo was designing a system for remotely monitoring wireless sensor networks through the Internet, and integrating all the data into business processes in order to execute data-mining operations and determine correlations among data. Scenarios related to "health at home" and remote monitoring of industrial facilities were designed in order to test the resulting system.

As an additional outcome, a mash-up of sensor collected data and Google Maps was developed, called SensorMap. This subsystem illustrated the potential applications of publishing up-to-date sensor data on the Internet in order to dynamically integrate different sources of information for providing higher level services.

Figure 1 shows a sensorized chair with two pressure sensors (at the seat and at the back) connected to a wireless sensor network node in order to provide information about its use to local and remote objects. Figure 2 contains a screenshot of the SensorMap application showing the state of the chair (if someone is sat on and/or leaned backwards).

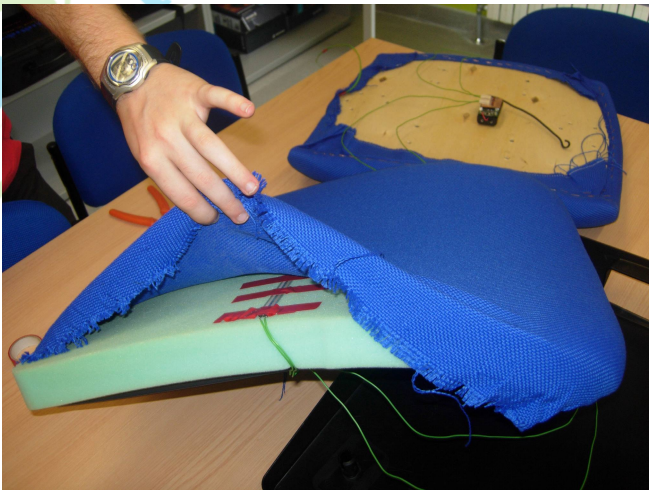


Fig. 1. Preparing the chair with pressure sensors and a WSN node

Flexeo's architecture was divided in to three different layers: the sensor layer, the concentrator layer and the business logic level.

The sensor layer was based on a wireless sensor network formed by Crossbow Mica2 nodes embedded into everyday objects such as a chair, a container, a

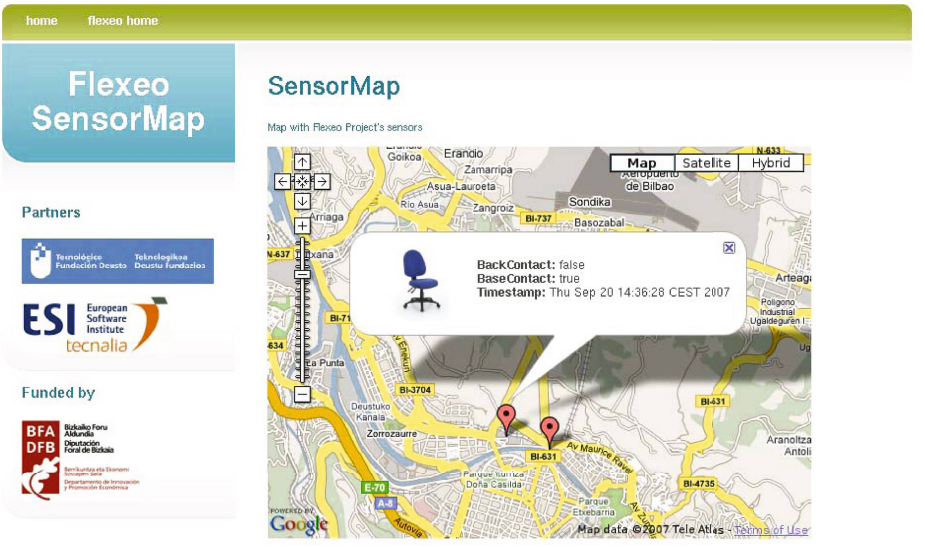


Fig. 2. Flexeo's SensorMap showing the state of a chair

wearable band and a wristwatch. The main goal was transforming different existing objects into wireless nodes, able to provide data about users' interactions.

The concentrator layer was implemented using an embedded computer, a Gumstix connex 400xm, connected both to the wireless sensor network and to the Internet via GPRS and/or Wi-Fi. The concentrator featured a middleware layer implemented using Java in the form of OSGi components, that was in charge of polling the sensor network, interpreting the data, evaluating rules in order to detect alarms, and grouping the data for sending them to the business logic servers that were on the Internet. In Flexeo, the concentrator was the architectural element in charge of connecting the sensor nodes, and thus, the real objects they were embedded in, to the Internet, acting as a communication gateway.

4.2 RealWidgets: From the Desktop to the Real World

Desktop-based operating systems are increasingly using some mini-applications called *widgets* or *gadgets* in order to provide small services or show concrete up-to-date information to users. Yahoo! Widgets, Apple Mac OS X Dashboard, Google Gadgets and Microsoft Windows Vista Gadgets are the most popular flavors of this form of interaction. Often, these widgets connect to online services in the Internet in order to provide weather or traffic information, most-popular YouTube videos, latests news, and so forth.

With RealWidgets we wanted to embody the functional capabilities of these digital entities into real world tiny wireless displays, in order to have small "windows" deployed everywhere opened to the information from the Internet.

The RealWidget is formed by an OLED display, with high resolution and contrast while small energy consumption, integrated with a Crossbow Mote2 wireless sensor network node. A computer acted as a gateway between the Internet and the wireless sensor network, running a RealWidget Management Application that connected to the required sites on the Web, downloaded the information, analyzed it and finally sent the content to the appropriate widget as configured by the user. Figure 3 illustrates a RealWidget showing information about the liquid level in a remote chemical container that was monitored by a wireless sensor network over the Internet.

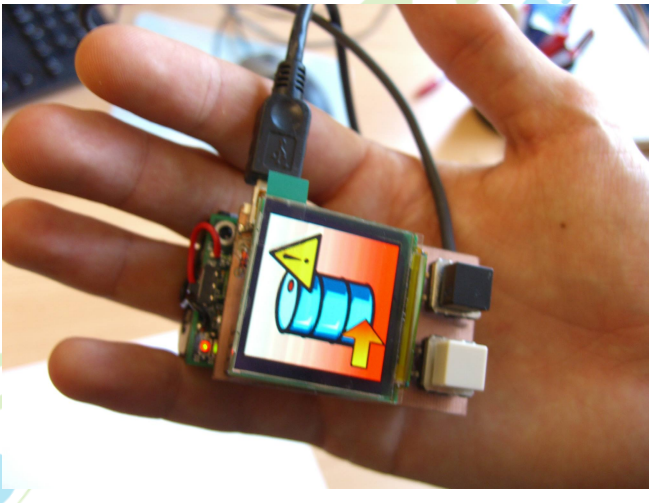


Fig. 3. RealWidget alerting about the liquid level in a tank

Two buttons were provided for interacting with the RealWidget, basically for managing the energy by postponing the information for a later time, or immediately discarding it.

One of the most popular widgets in computer desktops are those related to weather information, so one of the earliest examples of the RealWidget prototype involved obtaining the data from a weather information site on the Internet. We chose `weather.com` due to the availability of an open API that provided the ability to download an XML document with the weather forecast for any location in the world.

The RealWidget Management Application could host different adapters that acted as gateways between the Internet service and the physical RealWidgets. In the previous example, we created an adapter that was able to connect to the `weather.com` website, retrieve the XML document with the weather forecast of the configured location, and transformed it into a notification message that was sent to the RealWidget. If any change in the weather information forecast occurred, subsequent notification messages would be sent to the RealWidget. The Crossbow Mica2 wireless node on the RealWidget received the message,

processed it, and displayed an icon on the OLED display representing the weather conditions (sunny, cloudy, raining, and so forth).

4.3 SmartPlants: Autonomous Objects That Interact with Their Environment

SmartPlant was one of the evaluation prototypes we designed for the SoaM (Smart Objects Awareness and Adaptation Model) [21] platform: a completely decentralized architecture for designing semantic devices that collaborate by sharing information about the environment, and spontaneously react to changes in the context. The two pillars of SoaM are a semantic discovery protocol called mRDP (Multicast Resource Discovery Protocol) [19], and proactive semantic information exchange among entities. The result is an architecture in which devices create dynamic communication flows, and perform semantic reasoning over context information contributed by all the entities in the environment in order to determine the most appropriate behavior.

One of the scenarios we envisioned at the beginning of the research was to create an artifact that could be attached to real objects, augmenting their perceptions and providing them with intelligent capabilities. An additional challenge was to attach this kind of artifact to living entities, such as plants, in a way that could result in intelligent behavior carried out by the entities from the user's point of view.

An important part of this smart behavior was realized by obtaining additional knowledge about the environment from the Internet. Creating this kind of "smart plants" raised several new important implications such as:

- They could become first-class citizens in the environment, rather than passive elements. They could influence temperature, humidity or lighting settings.
- They could be perceived as autonomic systems [6] in a twofold view: as normal living beings they try to survive and adapt to environmental conditions; but also, as augmented intelligent entities they can interact and communicate with surrounding objects to create a more suitable and healthy environment.

Figure 4 depicts the smart plant prototype. The plant communicated at a semantic level with a wireless sensor network that provided temperature and light measures about different nearby locations, asking the user to me moved to the most suitable place using a synthesized voice. The plant was also able to download ontologies from the Internet in order to interpret particular expressions and predicates provided by surrounding sensors.

For all the semantic devices in SoaM, including the SmartPlant, we designed a computing platform consisting on a Gumstix connex 400xm with Wi-Fi communication capabilities, and a semantic middleware layer. The location a concrete ontology had to be downloaded from was stored in a central server on the Internet that acted as ontology index for the deployed objects.

In order to process semantic information and ontologies, we implemented a minimal semantic engine in the Gumstix called MiniOWL, able to interpret

Fig. 4. The SmartPlant prototype

the most popular ontological relationships such as `rdfs:subClassOf`, `owl:sameAs`, `owl:TransitiveProperty`, `owl:SymmetricProperty` and `owl:inverseOf` [18]. Festival Lite was used as text-to-speech engine in the embedded computer.

The information captured by the wireless sensor network nodes was semantized by a software component running on a computer connected to the sensor network. We designed specific ontologies using OWL for this scenario in order to represent knowledge about lighting conditions, temperature, and location.

4.4 Aware-Umbrella: A Reactive Device Integrating Local and Global Communication

The Aware-Umbrella was also a semantic device designed for the evaluation of the SoaM architecture. As already mentioned, the ability to seamless integrate local and global information sources in order to augment local intelligence and knowledge is of foremost importance for social devices. The most probable, but not unique, source for this information is the Internet, and particularly, available dynamic web services.

Our goal in this scenario was to design some kind of social object that could be aware of both environment-provided and Internet-provided information in order to take decisions and look more intelligent from users' perspective.

Our choice was to create a smart umbrella that could obtain current weather information from surrounding sensors, as well as the weather forecast for the next hours from the Internet. The smart umbrella reacted when the user was leaving home without taking it by issuing a synthesized voice alert (see Figure 5).

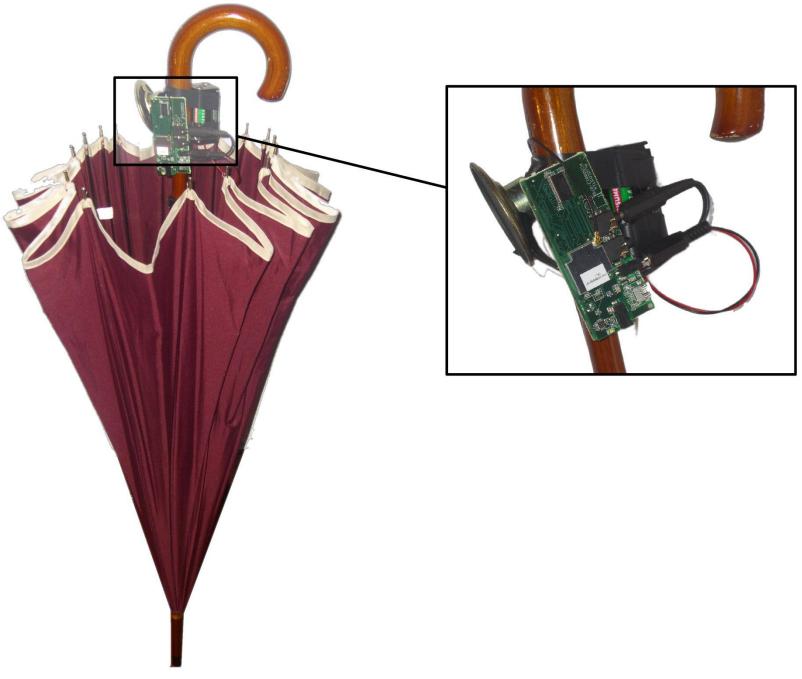


Fig. 5. The Aware-Umbrella prototype

The umbrella obtained context information from the Internet through a "virtual software sensor", a small piece of code that connected to the Internet to get weather information about the town (provided by `weather.com`) and semantized these data using a weather ontology we designed specifically for this case. The umbrella finally checked the state of the door in case the user was leaving when raining in order to decide whether to issue the voice alert.

In this scenario we used the same computing platform designed for the semantic devices in the SoaM project: a Gumstix connex 400xm with a Wi-Fi interface, a semantic middleware layer, and the Festival Lite TTS engine. The state of the door was provided by a proximity sensor (based on a magnetometer) on a Crossbow Mica2 at the door. As in the previous scenario, a computer acted as a gateway between the sensor network and the Wi-Fi network, semantizing the information provided by the node at the door.

5 Challenges for Web 2.0-Enabled Things

Web 2.0 is a synonym for collaboration, a natural habitat for the prosumer model (consumer + producer). Since web technologies have been the most popular infrastructure for designing objects that communicate on the Internet, are Web 2.0-enabled objects the next evolution of this paradigm? What are the interfaces and synergies between users participating in social networks and the Internet of Things?

Based on the experiences we have mentioned in the previous section, we have identified three main challenges on the integration of Web 2.0 collaborative models and Internet-enabled objects:

- Creation of social networks formed by collaborative devices
- Objects as consumers and producers of content in existing human-oriented social networks
- Objects as interfaces with Web 2.0-based services

Figure 6 depicts in a graphical way the relationships among these three challenges: devices collaborating in environment-specific social networks, information flows between devices and social websites, and objects as interfaces for people to interact with human-oriented social networks. In the next subsections, some discussion about these challenges is provided.

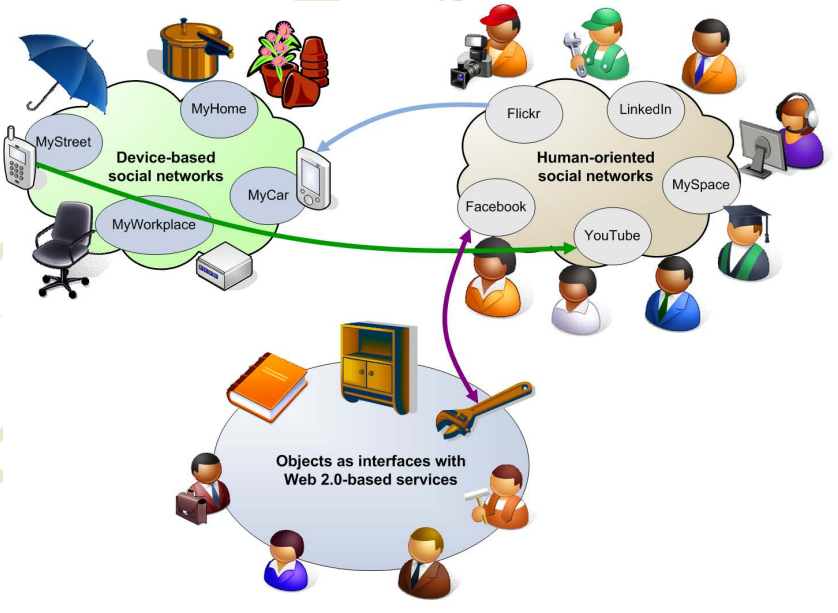


Fig. 6. Graphical representation of the three challenges for Web 2.0-enabled things

5.1 Creation of Social Networks Formed by Collaborative Devices

There are pervasive examples of Web 2.0 collaboration models that have boosted people participation: blogs, wikis, social tagging, social voting, content syndication, and so on. Are these models also suitable for boosting device collaboration on the Internet?

Devices can embrace Web 2.0-based communication models in order to implement a more human-like level of socialization. For people is relatively easy to understand the meaning of a blog entry, a photograph, or tags related to a

website, since semantic interpretation is inherently human. However, when it comes to applying these social models to the Internet of Things, the problem of automatic content interpretation appears as a major technical challenge.

In our experiments we found Semantic Web technologies, mainly RDF and OWL, a suitable means for objects to express their knowledge of the environment, enabling fellow objects to interpret this information and behave accordingly. There is no limit in the number of vocabularies than can be applied, depending on the knowledge domains objects must be aware of: semantic technologies are the appropriate vehicle to interpret the information provided by different heterogeneous sources.

For instance, in the Aware-Umbrella prototype, the "virtual weather sensor" that connected to the `weather.com` website semantized the weather information using a vocabulary the umbrella was able to interpret. This semantized weather information was made available to any object in the environment upon request. A similar model was followed in the SmartPlants prototype where a software component semantized the information provided by the wireless sensor network using lighting, temperature and location ontologies, and made it available to interested parties upon request.

While the SoaM architecture used in these scenarios promotes social collaboration among devices in an *ad hoc* way, Web 2.0 models fit perfectly here in order to obtain similar results in a more structured form. Instead of having the objects in the environment creating multiple communication flows to exchange information, an "environmental wiki" could act as repository of information about the environment, with multiple contributors that continuously update this wiki based on their perceptions and interpretations. Annotations in the wiki must still be made using Semantic Web technologies, which makes them difficult for humans to read, but very easy for devices to interpret.

There are two major approaches for the implementation of this "environmental wiki": a centralized approach based on a server in the environment with the semantic wiki software storing all the information; and a decentralized approach based on the wiki being stored in a distributed way, different parts in different objects with some kind of distributed searching mechanism. The later approach is similar to Tuple Spaces [14] or even Semantic Tuple Spaces [8], but with a wiki orientation.

Similarly, existing objects and devices generally maintain a log file with all the incidences that occur during normal operation. Access to this log file is generally restricted to administrators or special users. Just as people have transitioned from secret diaries to public blogs, in which users want to share their experiences, knowledge and opinions with everyone, we think that entities in the Internet of Things should follow a similar approach: creating blog entries using Semantic Web technologies so that objects can share their "personal" interpretations and knowledge. Microblogging (Twitter- and Jaiku- style) may be even more appropriate due to its brevity.

For example, we are working on an extension of the SmartPlant in which the plant creates blog entries in both human-readable and semantically annotated

form, in order to share *feelings* such as "I feel lonely", "I am thirsty", "I need more light", or "It is dark and nobody is here", based on its interpretation of the environmental information provided by surrounding entities.

5.2 Objects as Consumers and Producers of Content in Existing Human-Oriented Social Networks

The second challenge for Web 2.0-enabled things consists on connecting them to existing social networks in order to consume and/or produce content in behalf of their users. This approach would result in new ways of interacting with social websites, different from the traditional method of using the web browser.

For example, we are working on an additional extension to the SmartPlant in which the plant projects a blinking green light everytime the user has a Facebook friend request waiting in the queue. Thus, instead of having to check the social network homepage every now and then, existing objects in the environment can perform the task for their users and react in different ways.

Since the RealWidget prototype is basically a wireless OLED display, we are also working on a particularization of this device based on connecting to `Flickr.com` using the user's account, and start a photo slideshow everytime there are new photos in any of the user's groups.

While people are more and more involved in social networking on the Internet, these examples illustrate the powerful capabilities of using physical objects to relieve users from continuously checking up-to-date information from these websites. Designing things that actively contribute with content in behalf of the user seems a bit more complicated, but equally feasible, specially for simple activities.

For instance in the SmartPlant prototype, the user could automatically accept the pending Facebook friend requests by simply touching a part of the plant's pot where a small touch sensor resides. While we are not planning initially to augment the SmartPlant with a small display, it can be useful in this situation so that the user can visualize the name of the friend and accept or deny the request at his will. Precisely, user interactions with social websites-connected things is the subject of the third challenge we have identified for future research.

5.3 Objects as Interfaces with Web 2.0-Based Services

The major concern here is how to map user-object interactions to concrete actions on the social network. In the SmartPlant extension, is a green blinking projected light the most suitable means for alerting the user that he has some friend requests waiting? Is touching the pot the most appropriate means for indicating that the user accepts the requests?

Of course, a small display and keyboard in every object of the environment would eliminate any ambiguity about user interaction, but would also result in unnatural objects. Perhaps the most daunting challenge in having things connected to information sources on the Internet, and particularly to social websites, is how to map the information onto the object in a way that a user can intuitively understand, and even more difficult, how can user's interactions on the object be mapped onto concrete actions back to the website.

In the early experiments we carried out with the SmartPlant Facebook extension, we found that the user has to, obviously, have learned the interpretation of the blinking light previously as there is no natural mapping with its function. While this may be a problem if some concrete object has to express a plethora of different status or information updates using few interaction means, it is not such if there is only one simple function mapped onto the object as in the case of the mentioned Facebook extension. We consider this issue the most critical factor of success in order to design everyday objects that act as connectors with the user social life (on the Internet) in a natural way.

6 Conclusion

Devices should talk and share information without any geographical limitation, just as we humans do. Internet and the Web have demonstrated their ability for providing these capabilities to people, and also constitute the suitable means for enabling global-scale social devices.

This new wave of artifacts will exhibit social capabilities, e.g. promoting information sharing, recording individual opinions and interpretations about the context that can be easily accessed by others, tagging others' resources, or implementing voting and ranking strategies among themselves and the services they provide. As we described in this paper, Web 2.0 mechanisms are ideal candidates to be applied here.

We consider the concept of "social device" to be of foremost importance for creating smart environments. It comprises some of the fundamental aspects researchers have been looking for during the last years, especially intelligent and global context-awareness, and serendipitous collaboration even with remote objects and services.

In our research we investigated the foundations of this new wave of social objects that use the Internet for communication. We outlined their desired characteristics and designed experimental prototypes in several scenarios to demonstrate their functionality.

We also identified three main challenges in the research of social implications on the Internet of Things: the creation of social networks of devices in the environment based on semantic technologies, the connection of these objects with human oriented social websites, and the interaction challenges between virtual information and physical object characteristics.

This vision promotes the emergence of new scenarios in which surrounding objects become more interactive than ever before, being aware of the fact that the Internet is a natural extension of users' life.

Acknowledgments

This research has been partially supported by the Department of Innovation and Economic Promotion of the Biscay Regional Council (Spain), through the Ekinberri program funds for the project Flexeo. The authors would like to thank

all the research staff that took part in the projects Flexeo (especially Joseba Etxegarai, Xabier Laiseca and Pablo Orduña) and RealWidget (Iker Doamo), as well as Joseba Abaitua for his contributions on the social vision of devices.

References

1. Barton, J., Kindberg, T.: The cooltown user experience. Hewlett-Packard, Technical Report HPL-2001-22 (2001)
2. Chen, H., Finin, T., Joshi, A.: An intelligent broker for context-aware systems. In: Adjunct Proceedings of Ubicomp 2003, UbiComp, pp. 183–184 (October 2003)
3. Fielding, R.T., Gettys, J., Mogul, J.C., Frystyk, H., Masinter, L., Leach, P.J., Berners-Lee, T.: Hypertext Transfer Protocol – HTTP/1.1. IETF RFC 2616 (1999)
4. International Telecommuncations Union. The Internet of Things. Executive Summary (November 2005)
5. Issarny, V., Sacchetti, D., Tartanoglu, F., Sailhan, F., Chibout, R., Levy, N., Talamona, A.: Developing ambient intelligence systems: A solution based on web services. Automated Software Engineering 12(1), 101–137 (2005)
6. Kephart, J.O., Chess, D.M.: The vision of autonomic computing. Computer 36(1), 41–50 (2003)
7. Khare, R., Çelik, T.: Microformats: a pragmatic path to the semantic web. In: WWW 2006: Proceedings of the 15th international conference on World Wide Web, pp. 865–866. ACM Press, New York (2006)
8. Khushraj, D., Lassila, O., Finin, T.: Stuples: Semantic tuple spaces. In: Proceedings of Mobiquitous 2004: The First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, pp. 268–277 (2004)
9. Kindberg, T., Barton, J.: A web-based nomadic computing system. Computer Networks: The International Journal of Computer and Telecommunications Networking 35(4), 443–456 (2001)
10. Kindberg, T., Barton, J., Morgan, J., Becker, G., Caswell, D., Debaty, P., Gopal, G., Frid, M., Krishnan, V., Morris, H., Schettino, J., Serra, B., Spasojevic, M.: People, places, things: web presence for the real world. Mobile Networks and Applications 7(5), 365–376 (2002)
11. Lassila, O., Adler, M.: Semantic gadgets: Ubiquitous computing meets the semantic web. In: Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential, pp. 363–376 (2003)
12. de Ipiña, D.L., Vázquez, J.I., García, D., Fernández, J., García, I., Sainz, D., Almeida, A.: EMI^{2}lets: A Reflective Framework for Enabling AmI. Journal of Universal Computer Science 12(3), 297–314 (2006)
13. Masuoka, R., Labrou, Y., Parsia, B., Sirin, E.: Ontology-enabled pervasive computing applications. IEEE Intelligent Systems 18(5), 68–72 (2003)
14. Obreiter, P., Gräf, G.: Towards scalability in tuple spaces. In: SAC 2002, pp. 344–350. ACM Press, New York (2002)
15. O'Reilly, T.: What Is Web 2.0. Design Patterns and Business Models for the Next Generation of Software (September 2005, retrieved on January 2007), `http://www.oreillynet.com-/pub/a/oreilly/tim/news-/2005/09/30/-what-is-web-20.html`
16. Siorpaes, S., Broll, G., Paolucci, M., Rukzio, E., Hamard, J., Wagner, M., Schmidt, A.: Mobile interaction with the internet of things. In: 4th International Conference on Pervasive Computing (Pervasive 2006) (2006)

© IoT LAN Internet of Things

17. Song, Z., Masuoka, R., Agre, J., Labrou, Y.: Task computing for ubiquitous multimedia services. In: MUM 2004: Proceedings of the 3rd international conference on Mobile and ubiquitous multimedia, pp. 257–262. ACM Press, New York (2004)
18. Vazquez, J.I.: A Reactive Behavioural Model for Context-Aware Semantic Devices. PhD thesis, University of Deusto (2007)
19. Vazquez, J.I., de Ipiñato, D.L.: mRDP: An HTTP-based Lightweight Semantic Discovery Protocol. In: The International Journal of Computer and Telecommunications Networking ((to be published, 2007)
20. Vazquez, J.I., de Ipiña, D.L.: Principles and experiences on creating semantic devices. In: Proceedings of the 2nd International Symposium on Ubiquitous Computing and Ambient Intelligence - UCAMI 2007 (2007)
21. Vazquez, J.I., de Ipiña, D.L., Sedano, I.: SoaM: A Web-powered architecture for designing and deploying pervasive semantic devices. IJWIS - International Journal of Web Information Systems 2(3–4), 212–224 (2006)
22. World Wide Web Consortium. OWL Web Ontology Language Guide. World Wide Web Consortium, W3C Recommendation (February 2004)
23. World Wide Web Consortium. RDF Primer. World Wide Web Consortium, W3C Recommendation (February 2004)

Indoor Location Tracking Using Inertial Navigation Sensors and Radio Beacons

Pedro Coronel*, Simeon Furrer**, Wolfgang Schott, and Beat Weiss

IBM Research GmbH, Zurich Research Laboratory, Zurich, Switzerland
sct@zurich.ibm.com

Abstract. We consider the problem of real-time sensing and tracking the location of a moving cart in an indoor environment. To this end, we propose to combine position information obtained from an inertial navigation system (INS) and a short-range wireless reference system that can be embedded into a future"network of things". The data produced by the INS lead to accurate short-term position estimates, but due to the drifts inherent to the system, these estimates perform loosely after some time. To solve this problem, we also generate estimates with a wireless reference system. These radio-based estimates can be used as accurate long-term position estimates because their accuracy improves over time as the channel fading can be averaged out. We use a data fusion algorithm based on Kalman filtering with forward/backward smoothing to optimally combine the short- and long-term position estimates. We have implemented this localization system in a real-time testbed. The measurement results, which we obtained using the proposed method, show considerable improvements in accuracy of the location estimates.

1 Introduction

Real-time position localization of moving objects in an indoor environment is an important enabling technology for realizing the vision of the"internet of things" and will thus create numerous novel location-aware services and applications in various market segments. In the retail industry, for example, shopping carts equipped with shopping assistants enriched with additional location sensing means can guide customers through the store, provide them with location-based product information, and alert them to promotions and personalized discounts as they walk through the aisles. However, customer satisfaction, and thereby the success of this advanced shopping service, depends crucially on the achievable position accuracy. Moreover, the price of the shopping-cart tracking solution should be low enough to keep its attractiveness in the eyes of retailers.

Location tracking in an indoor environment can be done with various techniques based on mechanical, acoustical, ultra-sonic, optical, infrared, inertial, or

* Pedro Coronel is now with the Communication Technology Laboratory, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland.

** Simeon Furrer is now with Broadcom Corporation, Sunnyvale, CA, USA.

C. Floerkemeier et al. (Eds.): IOT 2008, LNCS 4952, pp. 325–340, 2008.
© Springer-Verlag Berlin Heidelberg 2008

radio-signal measurements. Localization algorithms in radio-based systems are frequently implemented measuring signal strength [1], angle of arrival, or time of arrival. As the radio beacons are deployed at known positions, triangulation or a signature method yield the location of the moving cart. As far as the radio technology is concerned, the short-range wireless personal-area and sensor network technology as specified by the IEEE 802.15.x standardization body and the ZigBee alliance are well-suited for obtaining low-cost implementations. However, the IEEE 802.11.x WLAN technology can also be attractive in buildings where an infrastructure is already deployed. The location estimates obtained with a radio-based system are stable on the long-term as they are updated rather infrequently, but they suffer from a large error variance due to channel fading.

On the other hand, location estimates can also be derived using inertial navigation sensors. Inertial measurements obtained with accelerometers, gyroscopes, and a compass are frequently applied in outdoor environments - often in conjunction with the global positioning system (GPS) - to track maneuverable vehicles like land crafts, aircrafts and ships [2]. Nevertheless, measurements obtained from an Inertial Navigation System (INS) can also be used for indoor tracking as they yield reliable short-term location estimates. Unfortunately, the localization algorithm has to be implemented with integrators and thus suffers from drifts that prevent keeping this accuracy over the long-term.

Various studies have shown that neither an inertial navigation system nor a radio-based positioning system on their own can achieve the target accuracy often required in an indoor environment. Therefore, we propose to merge both methods using a data fusion algorithm. This task can be solved in an optimal way by modeling the generation process of acceleration, bias, velocity, and position with a common state-space model and, posteriorly, estimating the position from measured data using an extended Kalman-filter algorithm that combines the short-term estimates produced by the INS and long-term estimates obtained from the radio positioning system. Fig. 1 shows a more detailed view of the three components of this new tracking system: the inertial navigation system, the wireless reference positioning system, and the data fusion system called location position estimator. In the sequel, we shall describe in detail each of these components and assess their localization accuracy performance by means of measurement results.

2 System Model

As shown in Fig. 1, the INS comprises an inertial measurement unit (IMU) with accelerometers and gyroscopes that measure acceleration and angular velocity in a right-handed Cartesian coordinate system (x, y, z), and a coordinate transformer that estimates the angle and rotates the coordinates system. The INS is mounted on a slowly moving cart in order to track its location. We assume that the z-axis is pointing in the opposite direction of the gravity vector; therefore, any movement of the cart on a plane floor can be tracked with one gyroscope and two accelerometers.

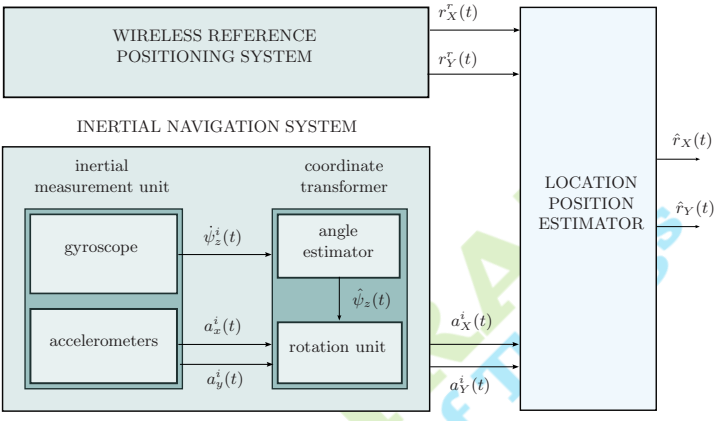


Fig. 1. Block-diagram of the new location tracking system. It comprises a wireless reference positioning system to produce long-term position estimates, an inertial navigation system that yields short-term estimates, and a location position estimator that optimally combines both using Kalman filtering.

2.1 Accelerometers

The accelerometers provide real-time measurements $a^i_x(t)$ and $a^i_y(t)$ on the acceleration of the cart in the x- and y-direction. Since these measurements are rather noisy due to cart vibrations, and often biased by a non-negligible, time-varying offset, we assume that each measurement $a^i(t)$ is related to the true acceleration $a(t)$ and the bias $b(t)$ according to

$$a^i(t) = a(t) + b(t) + v_a(t) \ , \tag{1}$$

where $v_a(t)$ is additive white Gaussian noise (AWGN) with a variance $\sigma^2_{v_a}$.

2.2 Gyroscope, Angle Estimator, and Rotation Unit

The gyroscope measures the angular velocity $\dot{\psi}^i_z(t)$ around the z-axis. Based on this noisy measurement, the angle estimator continuously estimates the rotation angle $\hat{\psi}^i_z(t)$. In a low-noise environment, this operation can be performed with an integrator. Its output value has to be initialized so that the x- and y-axis are aligned to the target coordinate system. After this calibration step, the rotation unit continuously transforms the measurements $a^i_x(t)$ and $a^i_y(t)$ to the target co-ordinate system by

$$\begin{bmatrix} a^i_X(t) \\ a^i_Y(t) \end{bmatrix} = \begin{bmatrix} \cos(\hat{\psi}^i_z) & -\sin(\hat{\psi}^i_z) \\ \sin(\hat{\psi}^i_z) & \cos(\hat{\psi}^i_z) \end{bmatrix} \begin{bmatrix} a^i_x(t) \\ a^i_y(t) \end{bmatrix} \ . \tag{2}$$

2.3 Process and Measurement Models

The acceleration $a(t)$ of the cart along one axis of the coordinate system is modeled by a first-order Markov process defined by

$$\dot{a}(t) = -\alpha a(t) + w_a(t) \ , \quad \alpha \geq 0 \ , \tag{3}$$

where $w_a(t)$ represents white Gaussian noise with variance $\sigma^2_{w_a}$, α denotes the correlation between successive acceleration values, and, hence, it is inversely proportional to the time maneuver constant of the cart. Different α values can be used for modeling the acceleration in distinct directions. Similarly, a time-varying bias $b(t)$ introduced by sensor imperfections can be modeled by

$$\dot{b}(t) = -\beta b(t) + w_b(t) \ , \quad \beta \geq 0 \ , \tag{4}$$

where $w_b(t)$ is AWGN with variance $\sigma^2_{w_b}$. The correlation coefficient β, however, takes on a larger value than α because the bias $b(t)$ changes with a much slower rate than the acceleration $a(t)$. The position $r(t)$ of the cart relates to its acceleration $a(t)$ according to

$$\ddot{r}(t) = \dot{v}(t) = a(t) \ , \tag{5}$$

where $v(t)$ denotes the cart velocity.

The linear differential equations (3)-(5) represent the continuous-time process model of the investigated system and can be written in the state-space form

$$\dot{\mathbf{x}}(t) = \mathbf{F}\mathbf{x}(t) + \mathbf{G}\mathbf{w}(t) \ , \tag{6}$$

where the state vector $\mathbf{x}(t)$, the process noise $\mathbf{w}(t)$, and the corresponding matrices $\mathbf{F}$ and $\mathbf{G}$ are given by

$$\mathbf{x}(t) = \begin{bmatrix} r(t) \\ v(t) \\ a(t) \\ b(t) \end{bmatrix} \ , \quad \mathbf{F} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -\alpha & 0 \\ 0 & 0 & 0 & -\beta \end{bmatrix} \ ,$$
$$\mathbf{w}(t) = \begin{bmatrix} w_a(t) \\ w_b(t) \end{bmatrix} , \quad \mathbf{G} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \ . \tag{7}$$

The measurement model is chosen in accordance with (1) and (2), and can be written as

$$\mathbf{z}(t) = \mathbf{H}\mathbf{x}(t) + \mathbf{v}(t) \ , \tag{8}$$

where the output vector $\mathbf{z}(t)$, the measurement noise $\mathbf{v}(t)$, and the matrix $\mathbf{H}$ are respectively given by

IoT IRAN Internet of Things

$$\mathbf{z}(t) = \begin{bmatrix} a^i(t) \\ r^r(t) \end{bmatrix} \quad , \quad \mathbf{H} = \begin{bmatrix} 0\,0\,1\,1 \\ 1\,0\,0\,0 \end{bmatrix} \quad ,$$
$$\mathbf{v}(t) = \begin{bmatrix} v_a(t) \\ v_r(t) \end{bmatrix} \quad . \tag{9}$$

The process and measurement model given by (7), (8) and (9) reflects the behaviour of the system if the cart is moving and location estimates are provided by the reference positioning system. If the cart is standing still, the dynamics are better described by incorporating the constraint $a(t) = v(t) = 0$ into the system matrix $\mathbf{F}$ of the process model (7). Similarly, if reference location estimates are not available, the matrix $\mathbf{H}$ in the measurement model (8) can be modified so that the signal $r^r(t)$ in the output vector $\mathbf{z}(t)$ is zero.

3 Wireless Reference Positioning System

The wireless reference positioning system provides estimates $r^r_X(t)$ and $r^r_Y(t)$ on the current position of the slowly moving cart in the target coordinate system (X, Y) using received signal strength indicators (RSSIs). For that purpose, we consider a wireless reference system, where K beacons (e.g. sensors equipped with a low-cost radio transmitter) are deployed at known positions over the location tracking area. Each beacon periodically broadcasts a radio packet containing the beacon identifier $k = 1, \ldots, K$. On the mobile cart, a radio receiver is mounted which continuously listens for incoming packets. When it receives a packet, the receiver determines the RSSI and decodes the packet to retrieve the originator's identifier k. The received signal strength indicator is used to approximate the distance d_k of the mobile to the corresponding beacon by $\hat{d}_k$. When the mobile has decoded radio packets from $K' \in \{3; K\}$ different beacons, the position of the mobile is given as the intersection of K' spheres centered on the packet originator position with radius $\hat{d}_k$. Various efficient methods can be employed to compute the position estimate on the basis of approximate distances [3]. As we assume that the cart is moving on the plane (X, Y), the computation takes into account the fact that the intersection lies in the plane $Z = 0$.

Next we shall discuss how to approximate the distance d_k on the basis of an estimate of the average RSSI derived from several decoded packets. The signal power received from the k^{th} beacon, RSSI_k, is related to d_k by a path-loss model, i.e. a random function that describes the signal attenuation as a function of the propagation distance, whose model parameters are specific to the tracking environment. Before describing the RSSI estimation algorithm, we discuss the path-loss model and how to determine the model parameters in a specific tracking environment.

3.1 Path-Loss Model

The mean signal attenuation described by the path-loss (PL) model decays exponentially with distance. However, the signal strength varies considerably around

its mean value due to the constructive and destructive superposition of multiple propagation paths between a beacon and the mobile. In an indoor environment, this effect is particularly important due to a typically obstructed line of sight, and reflections on walls and other objects (tables, shelves, window panes, ...). A general PL model that has been widely validated by experimental data [4] is given by

$$\mathsf{RSSI}(d) = \mathsf{RSSI}_0 + \alpha\ 10\log_{10}\left(\frac{d}{d_0}\right) + X_\sigma\ , \tag{10}$$

where RSSI_0 is an RSSI value measured in dB at a reference distance d_0 in meters (close to the transmitter) and α is the path-loss exponent. The random variable $X_\sigma \sim \mathcal{N}(0, \sigma^2)$ accounts for log-normal shadowing. More specifically, X_σ models the fact that measured signal levels at the same transmitter-receiver separation have a normal distribution around a distance-specific mean.

On the basis of this model, the transmitter-receiver separation can be approximated as follows: Suppose that the mobile successfully decodes at least one packet while it stands at d meters from the originating beacon, and it computes an estimate of the signal strength denoted by RSSI. Then, d can be approximated as

$$\hat{d} = d_0\ 10^{\frac{1}{10\alpha}(\mathsf{RSSI}(d) - \mathsf{RSSI}_0)}\ , \tag{11}$$

where we have used (10) and the model parameters tailored to the tracking environment.

3.2 Model Parameters

Depending on the environment, the parameters RSSI_0, α and σ in (10) take on different values (we assume that $d_0 = 1$ meter). For the purpose of identifying their values in a given environment, a large number of RSSI values are collected for multiple measurement locations *and* transmitter-receiver separations. The optimal parameters $\mathsf{RSSI}_0^\star$ and $\alpha^\star$ minimize the mean square error (MSE) between the measured data and the predicted values, that is

$$\arg\min_{\mathsf{RSSI}_0,\alpha} \sum_{n,m_n} \left(\mathsf{RSSI}_{m_n}(d_n) - \mathsf{RSSI}_0 - 10\alpha\log_{10}\left(\frac{d_n}{d_0}\right)\right)^2\ , \tag{12}$$

where m_n is an index that runs over the RSSI values collected at different positions for the same transmitter-receiver separation d_n. The parameter σ^2 describing log-normal shadowing is found as the variance of the error between the RSSI measurements and the values predicted by the model with parameters $\mathsf{RSSI}_0^\star$ and $\alpha^\star$.

3.3 RSSI Estimation

In this section, we describe a simple method to estimate the RSSI. Suppose that the time-varying wireless channel has the following input-output relation

$$y_n = h_n d^{-\frac{\alpha}{2}} x_n + z_n \ , \tag{13}$$

where y_n, x_n and h_n are respectively the received signal, transmit signal and channel gain at time n. We assume an ergodic (across time) fading process with channel gains h_n distributed according to $\mathcal{CN}(0,1)$ for each n. The noise z_n is a white Gaussian random process with spectral density N_0. Using N signal samples taken every T_s seconds, a signal power estimate can be computed as

$$P_N = \frac{1}{NT_s}\left(\sum_{n=1}^{N} |y_n|^2\right).$$

Upon application of the strong law of large numbers (SLLN), the sample mean converges to the statistical average with probability 1 as $N \to \infty$. As the channel and input signal are mutually uncorrelated, and so are the signal and noise components in (13), we have

$$\begin{aligned} P &= \lim_{N\to\infty} P_N \\ &\xrightarrow{\text{w.p.1}} \frac{1}{T_s}\left(d^{-\alpha}\mathbb{E}\left\{|h_n|^2\right\}\mathbb{E}\left\{|x_n|^2\right\} + \mathbb{E}\left\{|z_n|^2\right\}\right) \\ &= d^{-\alpha}\mathcal{P}_x + N_0 \ , \end{aligned} \tag{14}$$

where $\mathcal{P}_x$ is the mean signal power. Equation (14) suggests that d can be obtained from P by either using an estimate of SNR $= \frac{\mathcal{P}_x}{N_0}$, or by assuming that the noise power is negligible compared to the power of the signal component, i.e. $d^{-\alpha}\mathcal{P}_x \gg N_0$. In practice, P is approximated by P_N for a suitable N.

Note that the SLLN theorem does not only hold for i.i.d. random variables, but also for correlated variables - as the channel gains are - provided that the correlation vanishes for increasing lags [5]. In indoor environments, the Doppler shift is typically small and the wireless channel changes very slowly, i.e. large channel coherence time, implying that the number of samples N has to be carefully chosen depending on the channel coherence time and T_s to get reasonable distance estimates.

3.4 Implementation of the Reference Location Positioning System

We have implemented a reference location positioning system consisting of multiple Crossbow MicaZ motes [6], which comprise a Chipcon CC2420 RF transceiver attached to an Atmega128L micro-controller. The radio transceiver is compliant with the ZigBee/IEEE 802.15.4 standard [7]. It is thus operated in the 2.4 GHz ISM frequency band using direct sequence spread spectrum (DSSS) and can reach transmission data rates of 250kb/s. MicaZ motes are programmed in NesC as TinyOS applications to transmit or receive radio packets according to the application needs. Since MicaZ motes have been designed for a small size, low power consumption, and lost cost, they are well-suited for the use as pervasive devices.

In our testbed, most of the motes are deployed as radio beacons that transmit packets containing the beacon identifier k at a rate of 10 packets per second. One mote, however, is attached to the mobile cart to listen to incoming beacon transmissions. For every incoming packet, the mobile node first retrieves the originator identifier and the link quality information, and then forwards them through a serial interface to an attached portable computer. On the computer, we use a component object model to read the data forwarded by the mobile, before the data is processed online by MATLAB scripts.

In a MicaZ receiver, the signal is demodulated as illustrated in Fig. 2. After 10-bit A/D conversion and filtering, the frequency offset is compensated and an RSSI estimate is obtained by averaging the signal amplitude over 8 successive symbol periods corresponding to the standard-defined packet preamble having a duration of 128μs. In indoor environments, the channel does not change significantly during this time span, i.e., the channel coherence time is typically of the order of several milliseconds, and, therefore, we cannot assume (14) to hold for such a short averaging window. This suggests to average RSSIs across multiple packets to improve the distance estimate.

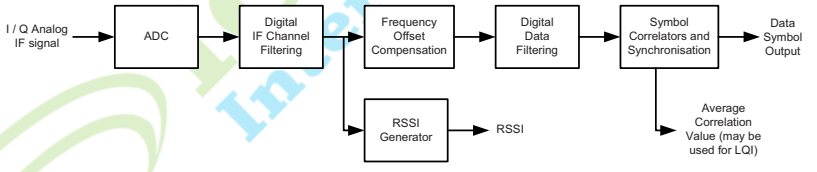


Fig. 2. Demodulator simplified block diagram [8]

Coming back to Fig. 2, we note that following digital filtering and before the data symbols can be recovered, synchronization is also performed based on the received packet preamble. The (peak) correlation value ρ obtained by matched-filtering the received signal with the preamble sequence is used to compute a link quality indicator (LQI). In general, the LQI is a function of both, the correlation value ρ and the RSSI. Using only the RSSI value as a LQI is not accurate enough as, for instance, a narrowband interferer inside the transmission bandwidth would increase the RSSI value although it actually reduces the true link quality. Hence, the correlation value provides a mean to determine the interference level in the received signal. Large correlation values correspond to low interference, while low correlation values evidence large interference. Corrupted RSSIs are thus be discarded as they will not lead to accurate distance estimates.

In summary, for each incoming packet, the mobile retrieves the originating beacon identifier k, the received signal strength RSSI_k value and correlation value ρ_k, and forwards them to the computer via the serial port. For $k = 1, 2, \ldots K'$, only the RSSI_k values corresponding to large correlation values ρ_k are used to approximate the distance d_k. When $K' \geq 3$, a position estimate is computed by triangulation.

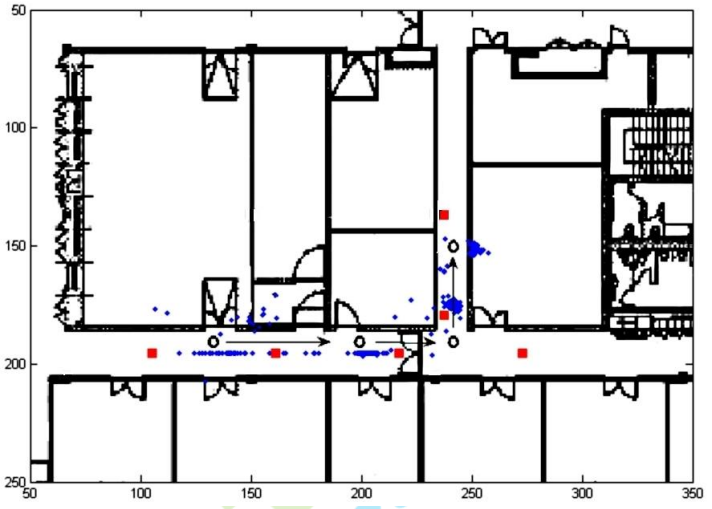


Fig. 3. Floor plan showing the beacon positions (red squares), the cart position at 4 stops (black circles) and the corresponding position estimates (blue dots). The dimensions are of approximately 30 x 20 meters.

3.5 Measurement Results

We have conducted several measurement campaigns in a commercial building to assess the performance of the proposed wireless reference positioning system. In a first measurement campaign, we derived the optimal PL model parameters for the given radio environment as described in Section 3.2. Then, we analyzed the accuracy of the position estimates obtained with the proposed system using the PL model with the optimized parameters.

Figure 3 shows the floor plan of the building where the measurements were performed. The arrows denote the path followed by the cart through the aisles of the building. The beacon positions are indicated by red squares. The mobile stops during 1 minute at each of the four different positions are shown by black circles in the figure. In blue, we plot the different position estimates derived from the RSSI values. Note that four dominating clouds can be distinguished close to the stops.

We are interested in evaluating two different alternatives to approximate d_k. The first method consists in generating a distance estimate $\hat{d}_k$ for each successfully decoded packet, i.e. the RSSI is computed over a time window of 128μs. The second method averages the RSSI_k for multiple incoming packets from beacon k as long as the mobile is standing still. Note that if the averaging window is large compared to the channel coherence time (i.e. several orders of magnitude), this will improve the convergence of the sample mean in (14); in turn, better estimates of d_k and more accurate position estimates will be obtained. This intuition is confirmed in Fig. 4, where we plot the cumulative distribution of the

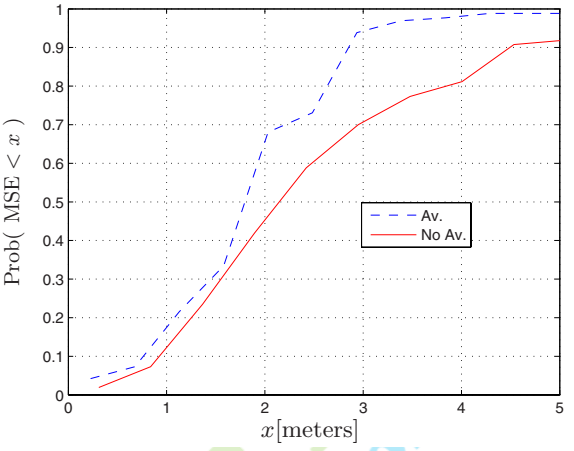


Fig. 4. Comparison of the cumulative distributions of the estimation error (MSE) with/whithout RSSI averaging

error obtained with and without RSSI averaging. Note that averaging results in a position error that is statistically smaller. For instance, the probability of a MSE smaller than 2 meters is about 0.45 without averaging and nearly 0.7 with averaging. When the RSSI values are averaged over multiple packets, the error is bounded to 3 meters in 95 % of the measurements, while this percentage drops to 70 % without averaging. In conclusion, computing an average RSSI using multiple packets improves significantly the accuracy of the position estimate as the sharp variations in the slow-fading channel can be averaged out.

4 Location Position Estimator

Short- and long-term position estimates obtained from the INS and the wireless reference system are combined using the location position estimator (see Fig. 1). The latter comprises a forward Kalman filter that yields optimal real-time location-position estimates $\hat{r}(t)$, a backward Kalman filter with a recording and smoothing unit for optimal smoothed estimates $\tilde{r}(t)$, a motion and reference signal detector, and a control unit for controlling the operation of the estimator. These components are illustrated in Fig. 5. Note that in this section the proposed location-positioning method and the corresponding algorithms are only given for one dimension of the target coordinate system (X, Y).

4.1 Forward Kalman Filter

The forward Kalman filter continuously monitors the noisy measured signals acceleration $a^i(t)$ and reference location $r^r(t)$, and computes in real-time a location-position estimate $\hat{r}(t)$ based on particular process and measurement

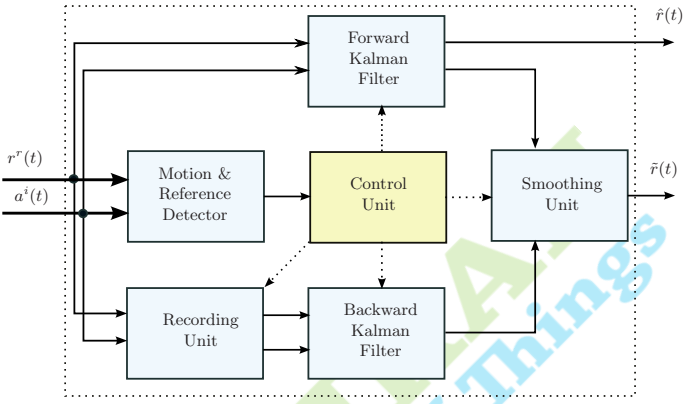


Fig. 5. Details on the various components of the location position estimator

models. This is achieved by, firstly, replicating the process and measurement model (without noise sources) in the filter to generate the state vector estimate $\hat{\mathbf{x}}(t)$ and the corresponding output vector $\hat{\mathbf{z}}(t) = \mathbf{H}\hat{\mathbf{x}}(t)$ and, secondly, continuously updating the state vector estimate according to

$$\dot{\hat{\mathbf{x}}}(t) = \mathbf{F}\hat{\mathbf{x}}(t) + \mathbf{K}(t)\left[\mathbf{z}(t) - \hat{\mathbf{z}}(t)\right] \quad . \tag{15}$$

$\hat{\mathbf{x}}(t)$ is an optimal estimate of $\mathbf{x}(t)$ in the mean square error sense. The filter is driven by the error between the measured signal vector $\mathbf{z}(t)$ and the reconstructed output vector $\hat{\mathbf{z}}(t)$ weighted by the Kalman gain matrix $\mathbf{K}(t)$. To obtain the optimum Kalman filter gain settings, the covariance matrix of the state vector estimation error also has to be computed as described, for example, in [9].

A realization of the Kalman filter obtained for the process and measurement model given in (6)-(9) comprises eight time-varying Kalman gains as shown in Fig. 6. The estimates for the location $\hat{r}(t)$, velocity $\hat{v}(t)$, and accelaration $\hat{a}(t)$ of the cart, and accelerometer bias $\hat{b}(t)$ are provided at the output of four integrators. Since three of the four integrators are cascaded, the filter is especially sensitive to DC-offsets present at the input of the integrators leading to severe drift effects in the integrator output signals. As discussed below, additional bias and drift compensation techniques are applied to cope with this inherent deficiency.

4.2 Backward Kalman Filter with Recording and Smoothing Unit

While the forward Kalman filter provides real-time estimates $\hat{\mathbf{x}}(t)$ on the basis of measurement signals obtained up to time t, a better estimate $\tilde{\mathbf{x}}(t)$ can be obtained by forward-backward Kalman filtering. Intuitively, this technique is able to improve the estimate by exploiting past and future data. A drawback is that the smoothed estimate $\tilde{\mathbf{x}}(t)$ can no longer be computed in real-time, but only after some delay. However, this inherent disadvantage can be dealt with

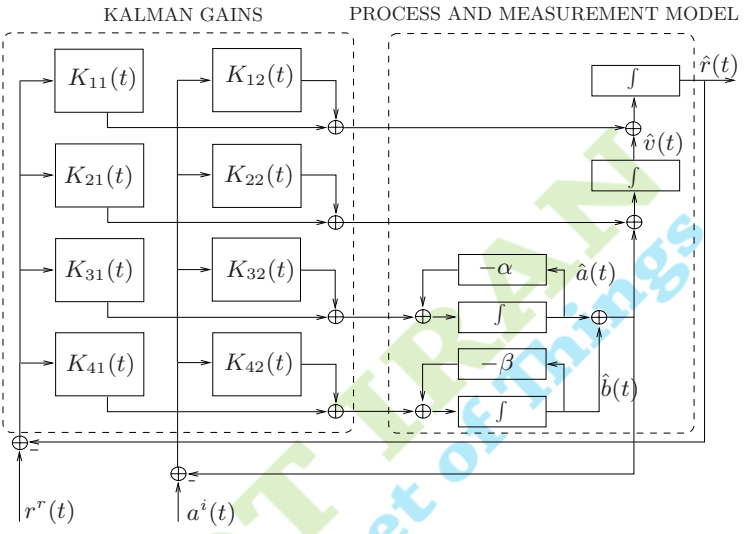


Fig. 6. Kalman Filter Designed for Moving Cart in Presence of Reference Signal

when the state vector does not change its value. Hence, we propose to apply forward-backward smoothing whenever the cart is not moving.

The forward-backward smoothing algorithm can be realized with a recording unit, a forward Kalman filter, a backward Kalman filter, and a smoothing unit as shown in Fig. 5. When a cart starts moving at any arbitrary time t_0, the recording unit starts storing the signals acceleration measurement $a^i(t)$ and reference location $r^r(t)$ until the cart stops at time t_1. At that moment, the location position estimator computes the smoothed estimate $\tilde{\mathbf{x}}(t)$ using forward-backward filtering over the data collected during the time interval $[t_0, t_1]$. The forward Kalman filter is implemented as discussed previously with the initial conditions $\hat{r}(t_0) = r_0$, $\hat{b}(t_0) = b_0$, $\hat{a}(t_0) = \hat{v}(t_0) = 0$. It runs forward in time from t_0 to t, and computes the forward state vector estimate denoted by $\hat{\mathbf{x}}^f(t)$. The backward Kalman filter has the same structure as the forward filter but it runs backward in time, i.e. from t_1 to t, and uses the following initial conditions: the accelaration $\hat{a}(t_1)$ and the velocity $\hat{v}(t_1)$ are set to zero because the cart is not moving, and the initial location $\hat{r}(t_1)$ and bias $\hat{b}(t_1)$ are modeled as Gaussian random variables. The estimate provided by the backward Kalman filter is denoted by $\hat{\mathbf{x}}^b(t)$.

Finally, the smoothing unit optimally combines the estimates $\hat{\mathbf{x}}^f(t)$ and $\hat{\mathbf{x}}^b(t)$ to generate $\tilde{\mathbf{x}}(t)$. Optimal combining is accomplished by weighting the estimates for each t using the covariance matrices of their corresponding state vector estimation error that are provided by the forward and backward filters.

4.3 Motion and Reference Signal Detector with Control Unit

The motion and reference signal detector continuously monitors the signals $a^i(t)$ and $r^r(t)$ for two purposes: firstly, to detect whether the cart is *in-motion* and,

secondly, to recognize the *presence-of-reference* data or the *absence-of-reference* data. The latter may also be directly signaled to the location position estimator by the reference positioning system.

The motion detector continuously traces the magnitude of the acceleration measurement signal $a^i(t)$. As it strongly oscillates when the cart is moving, we sample it at rate $\frac{1}{T}$ and average N consecutive samples to obtain the $\overline{a}(kT) = \frac{1}{N}\sum_{l=0}^{N-1}|a^i(kT-lT)|$. Then, an *in-motion* event is declared if

$$\overline{a}(kT) > \mu \ , \tag{16}$$

where μ is a pre-determined threshold. Both N and μ have to be chosen so as to reliably detect short cart stops, but also to ensure that false decisions are rather unlikely.

Depending on the events *in-motion/no-motion* and *presence/absence-of- reference*, the control unit dynamically switches the location position estimator between the following basic modes of operation:

1. **Cart *In-Motion* and *Presence-of-Reference* Signal.** The control unit enables the operation of the forward Kalman filter. The filter provides estimates of the location, acceleration, and velocity of the cart, as well as an estimate of the accelerometer bias in real time. Moreover, the acceleration $a^i(t)$, the reference location $r^r(t)$, and the mode parameter are recorded in a storage device for posterior use (mode 3).

2. **Cart *In-Motion* and *Absence-of-Reference* Signal.** Since no reference signal is available, the location position estimator derives an estimate based on the accelerometer measurements $a^i(t)$ only. This change is modeled by setting all elements in the second row of the measurement matrix $\mathbf{H}$ to zero. The control unit reconfigures the Kalman filter so as to reflect this change of the underlying measurement model in the filter. Moreover, the control unit freezes the bias estimate $\hat{b}(t)$ in the Kalman filter to its current value when the mode of operation is switched from *presence-of-reference* to *absence-of-reference* signal. This freeze is recommendable to avoid observability problems due to simultaneous changes of acceleration and bias in the system model while no reference signal is present. Finally, the acceleration $a^i(t)$, the reference location $r^r(t)$, and the mode parameter are recorded in a storage device for posterior use (mode 3).

3. ***No-Motion* of the Cart.** Whenever the motion detector detects a *no-motion* event, the control unit overrides to zero the acceleration estimate $\hat{a}(t)$ and the velocity estimate $\hat{v}(t)$ in the Kalman filter. This calibrates the integrator output signals to zero, and thus prevents offset accumulation by the integrators. Moreover, an enhanced process model for the dynamic behavior of the cart is obtained by incorporating the constraint $a(t) = v(t) = 0$ into the process equations (6) and (7). The control unit reconfigures the Kalman filter to reflect this change of the underlying process model in the filter.

Since the measurements $a^i(t)$ and $r^r(t)$ were recorded while the cart was *in-motion* (see mode 1), the data stored up to the latest stop of the cart can be post-processed by forward-backward smoothing. The control unit initializes the execution of the smoothing when the event *no-motion* is detected. After this operation, the location position estimator outputs the optimal smoothed estimate of the cart location.

4.4 Measurement Results

We have implemented the inertial navigation system and the location position estimator by mounting a commercially available inertial measurement unit (IMU) to a mobile cart and attaching the IMU to a portable computer carried by the cart. As IMU, we selected the sensor unit MT9-B from Xsens Motion Technologies [10], which can easily be embedded in a cart due to its small size and light weight. The unit measures its own acceleration in the x- and y-direction, as well as its angular rate around the z-axis. The sensed data are digitized at a sampling rate of 100 Hz and forwarded in real-time through a serial interface to the portable computer. On the computer, MATLAB scripts are executed to derive from the sensed data and reference signals the current location position of the cart using the Kalman filter algorithm introduced above. On the mobile

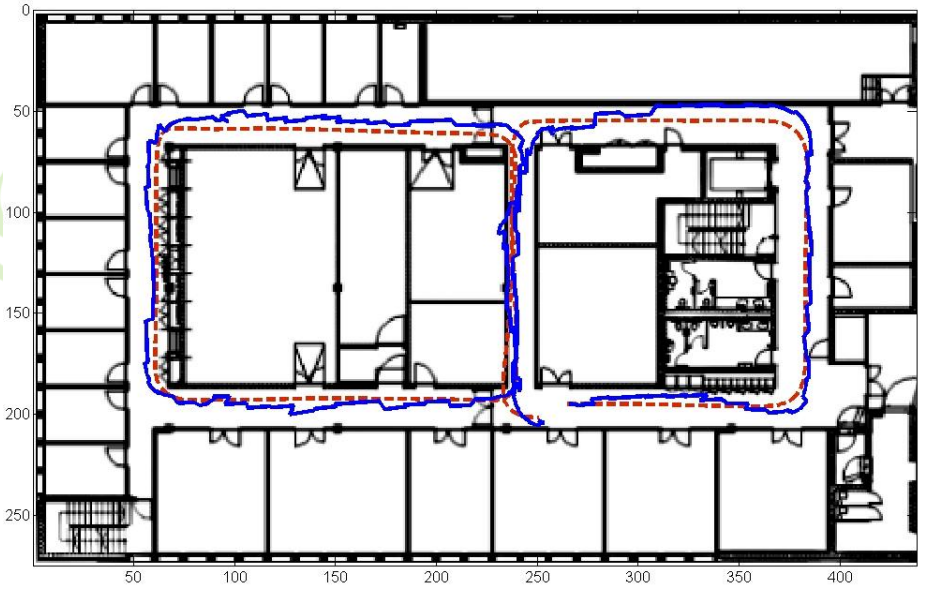


Fig. 7. Measurements conducted at an industrial building. The red dashed line represents the true path of the cart as indicated by the odometer; the blue solid line represents the position location estimates as indicated by the combination of the INS measurements and the reference signal using forward-backward Kalman filtering.

cart, we also mounted an odometer to track its true position and compare it to the estimates provided by the proposed location positioning system. With this prototype implementation of a location-aware mobile cart, a measurement campaign has been conducted in an industrial building to assess the performance of the location positioning system.

Fig. 7 shows the floor plan corresponding to the building where the measurements were performed. The mobile cart was moved in the aisles of the building along the red dashed path. This trajectory has been recorded with the odometer. The blue solid line indicates the location position estimates that have been obtained with the proposed location sensing and tracking solution. We observe that the Kalman filtering method presented by the blue trajectory compares favorably to the actual trajectory followed by the cart.

5 Conclusions

We have presented a novel indoor location tracking system that can be embedded into the envisioned "internet of things". The system uses an inertial navigation system and a wireless reference system for real-time sensing and tracking the location of a moving cart in an indoor environment. The wireless reference system uses multiple low-cost beacons deployed over the tracking area that periodically transmit radio packets. The signal strength of the received packets at the mobile is used to approximate the distance to its originator, and distances corresponding to multiple beacons are combined to determine the position of the mobile. We have used an experimental setup with MicaZ motes to demonstrate that a simple path-loss model and received signal power estimation algorithm can be used to derive a reference position with a reasonable accuracy. The PL model parameters have been derived in a preceding measurement campaign, before they have been used for online computation of the location estimates. By averaging the RSSI values over multiple packets, we obtained a more robust location estimate.

In order to improve the accuracy of the position estimate, we have proposed and investigated a merge of the measurements obtained by the INS and the reference positioning system. This is achieved by a data fusion algorithm that is based on Kalman filtering. The generation processes for the acceleration, bias, velocity and position are modeled using a common state-space model. The position estimates are obtained from measured data using an extended Kalman filter that exploits the reference position. Measurement results have shown the improvement that the proposed method is able to achieve.

References

1. Xiang, Z., Song, S., Chen, J., Wang, H., Huang, J., Gao, X.: A wireless LAN based indoor positioning technology. IBM Journal of Research and Development (2004)
2. Singer, R.A.: Estimating optimal tracking filter performance for maneuvering targets. IEEE Trans. on Aerospace and Electronic Systems (1970)

3. Murphy, W., Hereman, W.: Determination of a position in three dimensions using trilateration and approximate distances. Technical Report MCS-95-07, Mathematical and Computer Sciences Dept., Colorado School of Mines (1995)
4. Rappaport, T.S.: Wireless Communications. Prentice Hall, New Jersey (1996)
5. Serfling, R.J.: Approximation Theorems of Mathematical Statistics. In: Probability and Statistics, Wiley, Chichester (2002)
6. Crossbow Technology, Inc.: MPR-MIB User Manual
7. IEEE: Standard for information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements - Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs), IEEE, Los Alamitos (2003)
8. Texas Instruments, Inc.: Chipcon Products - 2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver
9. Maybeck, P.S.: Stochastic Models, Estimation and Control. Academic Press, New York (1982)
10. Xsens Technologies, B.V: MT9 Inertial 3D Motion Tracker

IOT IRAN Internet of Things

Tandem: A Context-Aware Method for Spontaneous Clustering of Dynamic Wireless Sensor Nodes

Raluca Marin-Perianu[1], Clemens Lombriser[2], Paul Havinga[1], Hans Scholten[1], and Gerhard Tröster[2]

[1] Pervasive Systems Group, University of Twente, The Netherlands
[2] Wearable Computing Lab, ETH Zürich

Abstract. Wireless sensor nodes attached to everyday objects and worn by people are able to collaborate and actively assist users in their activities. We propose a method through which wireless sensor nodes organize spontaneously into clusters based on a common context. Provided that the confidence of sharing a common context varies in time, the algorithm takes into account a window-based history of believes. We approximate the behaviour of the algorithm using a Markov chain model and we analyse theoretically the cluster stability. We compare the theoretical approximation with simulations, by making use of experimental results reported from field tests. We show the tradeoff between the time history necessary to achieve a certain stability and the responsiveness of the clustering algorithm.

1 Introduction

Wireless sensor networks, smart everyday objects and cooperative artefacts represent all different facets of the ubiquitous computing vision, where sensor-enabled devices become integrated in the environment and provide context-aware services to the users. Various systems have already been demonstrated to be able to retrieve the context, such as the physical context (e.g. position, movement [5]), the situation (e.g. meeting [17]) and even the emotional context (e.g. mood detection [4]). One step further is to have sensor nodes that interact and use common contextual information for reasoning and taking decisions at the point of action. Such a "networked world" opens perspectives for novel applications in numerous fields, including transport and logistics [10], industrial manufacturing [13], healthcare [8], civil security and disaster management [9].

In this paper, we explore a non-traditional networking paradigm based on context sharing. Previous work showed that sensor nodes can recognize online a common context and build associations of the type "moving together" [7,10]. Starting from this result, we address the problem of organizing the nodes sharing a common context into stable clusters, given the dynamics of the network and the accuracy of the context-recognition algorithm.

The contributions of this paper are as follows. Firstly, we propose Tandem, an algorithm for spontaneous clustering of mobile wireless sensor nodes. The

C. Floerkemeier et al. (Eds.): IOT 2008, LNCS 4952, pp. 341–359, 2008.
© Springer-Verlag Berlin Heidelberg 2008

algorithm allows reclustering in case of topological or contextual changes, and tries to achieve stable clusters if there are no changes in the network, by analysing the similarity of the context over a time history. Secondly, we approximate the behaviour of the algorithm using a Markov chain model, which allows us to estimate the percentage of time the clustering structure is correct given that the topology is stable. Thus, we are able to analyse the cluster stability both theoretically and through simulations, using experimental results reported from real field tests. Thirdly, we study the tradeoff between the time history necessary to achieve a certain stability and the responsiveness of the clustering algorihm. As a result, we estimate the delay induced by the time history, given a desired cluster stability.

2 Application Scenarios

We describe two applications where wireless sensor nodes are able to extract and communicate the general contextual information for creating a dynamic cluster. The clusters then provide services such as reporting the group membership, analysing the cluster activity, recognizing fine-grained events and actions.

2.1 Transport and Logistics

Transport and logistics represent large-scale processes that ensure the delivery of goods from producers to shops [10]. Using the wireless sensor networks technology in transport and logistics is particularly interesting for dynamically locating the goods, generating automatic packing lists, as well as for monitoring the storage condition of a product (e.g. temperature, light) or its surroundings.

The delivery process starts at a warehouse, where the transport company personnel assemble rolling containers (Returnable Transport Items - RTIs), pick the requested products from the warehouse shelves, and load them in the RTIs. Next, the RTIs are moved on the *expedition floor*, a large area used for temporary storage (see Figure 1). From the expedition floor, the RTIs are loaded into trailers according to the shop orders. In order to avoid the errors made during loading of items in the RTIs, a node equipped with a movement sensor can be attached to each product. At the moment the RTIs are pushed on the expedition floor, the sensors from each container correlate their movement and report as a group to the devices carried by the company personnel. In this way, a missing or wrong item can be detected before arriving on the expedition floor. The same solution can be applied for checking the correct loading of the RTIs into the trailers that deliver the goods to the shops.

In a similar manner works the list generator for automatic packing [1]. Various order items are packed in a box and an invoice is generated. In order to find out where a certain item is, nodes with movement sensors can be attached to each good. When the box is moved around, the items inside can correlate their movement and decide that they form a group. When the objects are grouped, checking on the goods and packing lists can be done automatically.

Fig. 1. Transport and logistics scenario

2.2 Body Area Networks (BAN)

Wearable computing aims at supporting workers or people in everyday life by delivering context-aware services. One important aspect is the recognition of human activities, which can be inferred from sensor integrated into garments and objects people are interacting with. The usage of context information enables a more natural interaction between humans and computers, and can assist workers with complex tasks with just now relevant information. Examples include training unskilled workers for assembly tasks [13], monitoring the health and activity of patients [8], assisting firefighters engaged in rescue operations in unknown environments with poor visibility.

Clustering of nodes based on the movement of persons simplifies the selection of relevant sensors in the environment which can contribute to the recognition of the currently performed activity. The advantage is that the recognition processing can be kept within the cluster, which is important for environments where multiple people are present. This provides: (1) identification of the body wearing the sensors, (2) a better recognition stability when selecting sensors moving with a person for the recognition task, and (3) potentially a trusted network where private data can be communicated only to nodes within the cluster.

3 Related Work

Clustering in ad-hoc and sensor networks is an effective technique for achieving prolonged network lifetime and scalability [14]. Parameters such as the node degree, transmission power, battery level, processor load or degree of dynamics usually serve as metrics for choosing the optimal clustering structure. Nevertheless, recent initiatives address the problem of grouping based on application-specific attributes. For instance, Bouhafs et al. [3] propose a semantic clustering algorithm for energy-efficient routing in wireless sensor networks. Nodes join the clusters depending on whether they satisfy a particular query inserted in the

network. The output of the algorithm is called a *semantic tree*, which allows for layered data aggregation.

The Smart-Its project [6] first introduces the notion of context sharing: two smart objects are associated by shaking them together. Using this explicit interaction between the two devices, an application-level connection can be established. For example, the two devices can authenticate using secret keys that are generated based on the movement data [11]. Siegemund [12] proposes a communication platform for smart objects that adapts the networking structure depending on the context. A clusterhead node decides which nodes can join the cluster, based on a similar symbolic location. Strohbach and Gellersen [15] propose an algorithm for grouping smart objects based on physical relationships. They use associations of the type "objects on the table" for constructing the clusters. A master node (the table) has to be able to detect the relationships for adding/deleting the nodes to/from the cluster. The solution described is suitable for static networks, where the master node is stationary for a long period of time. In our work, the network is dynamic, the context is permanently changing and every pair of nodes is capable of understanding the physical relationships, and thus the common context.

We now give two examples of shared-context recognition algorithms, where the practical results help us evaluate the clustering algorithm. First, Lester et al. [7] use accelerometer data to determine if two devices are carried by the same person. The authors use a coherence function to derive whether the two signals are correlated at a particular frequency. Second, in our previous work [10], we proposed a correlation algorithm which determine whether dynamic sensor nodes attached to vehicles on wheels move together. We use the real-world experimental results reported by both algorithms to analyse the performance of Tandem.

4 Algorithm Description

The goal of Tandem is to organize the nodes that share the same context, so that they can subsequently collaborate to provide a service. Tandem assumes that each node runs a *shared-context recognition algorithm*, which provides a number on a scale, representing the *confidence value* that two nodes are together. This algorithm can be for example a *coherence* function, which measures the extent to which two signals are linearly related at each frequency [7] (on a scale from 0 to 1), or the *correlation coefficient*, which indicates the strength and direction of a linear relationship [10] (on the scale from -1 to 1). Such an algorithm permanently evaluates the context, so that at each time step every node has an updated image of the current situation, reflected in a new set of confidence values (one for every neighbour). Since the context recognition algorithm has a certain accuracy, the perceived context determined by the confidence values may vary in time.

4.1 Requirements

Following the scenarios from Section 2, the nodes sharing the same context are within each-others transmission range, so we consider only one-hop clusters. The

environment is dynamic, where the topology and context can change at any time. The requirements for the clustering algorithm are the following:

1. *Incorporate dynamics.* The clusters can merge or split, depending on the context changes. Nodes can join and leave the cluster if the topology or context changes. For example, in the BAN scenario, people can pick up and use different tools, and then return or exchange them with other people. In this case, the nodes attached to the tools have to join and leave the BAN clusters. Contextual and topological changes cannot be predicted, so the algorithm cannot assume a stable situation during cluster formation.
2. *Stability.* If there are no contextual or topological changes, the clustering structure has to be stable. Following the remark that every node periodically re-evaluates the shared context with its neighbours, the fluctuations of the confidence values may lead to unwanted changes in the cluster structure. Therefore, the cluster has to cope with these fluctuations in order to keep the structure as stable as possible. A solution to increase the stability is to analyse the similarity of the context over a larger time history. In this sense, a tradeoff has to be found between the spontaneity in accommodating changes and the desired cluster stability.
3. *Energy-efficiency.* The communication overhead should be kept to a minimum, for prolonging the lifetime of the wireless network.
4. *Facilitate service provisioning.* The clusters have to easily interact with the higher-layer applications and provide context-aware services to the user.

4.2 Cluster Formation Algorithm

For the sake of simplicity, we assume that a node is aware of its neighbours (every node within the range) and the one-hop communication is reliable (this can be achieved for example by using a simple stop-and-wait ARQ protocol).

Each node v periodically computes the confidence of sharing the same context with its neighbours. If the confidence with a neighbour u exceeds a certain threshold, then v considers that it shares the same context with u for the given time step. The final decision for sharing the same context with u is founded on the confidence values from a number of previous time steps, called the *time history* (see Section 5.1).

The fact that the perception of the shared context may vary from one node to another leads to nodes having different views of the cluster membership. To provide a consistent management of the membership list and cluster organization, a *clusterhead* or *root* node is dynamically elected among the nodes that share the same context. This also assists the service provisioning, by having a single point of interaction with the higher-layer applications (see Requirement 4). In order to allow merging of clusters and to facilitate the election process, the candidate clusterheads dynamically generate unique *priority numbers*, either based on the hardware addresses, the capabilities of the nodes based on the resources available or as a context-dependant measure. A regular node subscribes to the clusterhead with which it shares a common context and has the highest priority number.

Algorithm 1. Tandem - node v (events/actions)

Initialization:
1. $r(v) \leftarrow \perp,\ r(u) \leftarrow \perp,\ \forall u \in \Gamma(v)$

GetContext:
1. $r_0(v) \leftarrow r(v)$ // Store the root of v
2. Update $h(v,m), \forall m \in \Gamma(v)$ // Update the history
3. $M \leftarrow \{m \in \Gamma(v) \mid h(v,m) > h_{min}\}$ // Select the nodes sharing the same context with v
4. **if** $M \neq \emptyset$ **then**
5. $M_0 \leftarrow \{ m \in M \mid r(m) = m\}$ // Select from M the nodes that are clusterheads
6. **if** $M_0 \neq \emptyset$ **then**
7. Choose $u \in M_0$ such that $pn(u) = max\{pn(m) \mid m \in M_0\}$
8. **if** $(r(v) = \perp) \vee (r(v) \notin M_0) \vee (r(v) = v \wedge pn(v) < pn(u))$ **then**
9. $r(v) \leftarrow u$ // Choose u as the root of v
10. **end if**
11. **else if** $r(v) \neq v$ **then**
12. $M_1 \leftarrow \{ m \in M \mid r(m) \neq \perp\}$ // Select from M the nodes that have a valid root
13. **if** $M_1 \neq \emptyset$ **then**
14. $r(v) \leftarrow \perp$
15. **else**
16. $r(v) \leftarrow v$ // Become root
17. Generate $pn(v) > 0$
18. **end if**
19. **end if**
20. **else**
21. $r(v) \leftarrow \perp$ // v is unassigned
22. **end if**
23. **if** $r_0(v) \neq r(v)$ **then**
24. Send *SetRoot* $(v, r(v), pn(v))$ to neighbours // Announce root change
25. **end if**

We use the following notation:

- V is the the set of nodes in the network.
- n is the number of nodes in the network, $n = |V|$.
- $r(v)$ is the root or clusterhead of node v.
- $pn(v)$ is the priority number of node v.
- $\Gamma(v)$ is the neighbourhood of node v.
- $h(v,u)$ represents the number of times v and u are sharing a common context over a total time history of H steps.
- h_{min} is the minimum amount of times when two nodes are sharing a common context, such that they can safely be considered part of the same cluster.

The algorithm constructs a set of one-hop clusters, based on the context information shared by the nodes. A node v can be: (1) *unassigned*, where v is not part of any cluster, (2) *root*, where v is clusterhead, or (3) *assigned*, where v is assigned to a cluster where the root node is one of its neighbours.

Algorithm 1 gives the detailed description of the cluster formation and update of knowledge among neighbouring nodes. Every node has the following information about its neighbours: the root, the priority number and whether it shares a common context for a specified time history. Let v be an arbitrary node in the network. At each time step, node v changes or chooses its root node in the following cases: (1) v is unassigned, (2) v does not share a common context with its root, (3) the root of v is no longer a root or (4) v is root and there is another neighbour root, sharing the same context with v, that has a higher priority

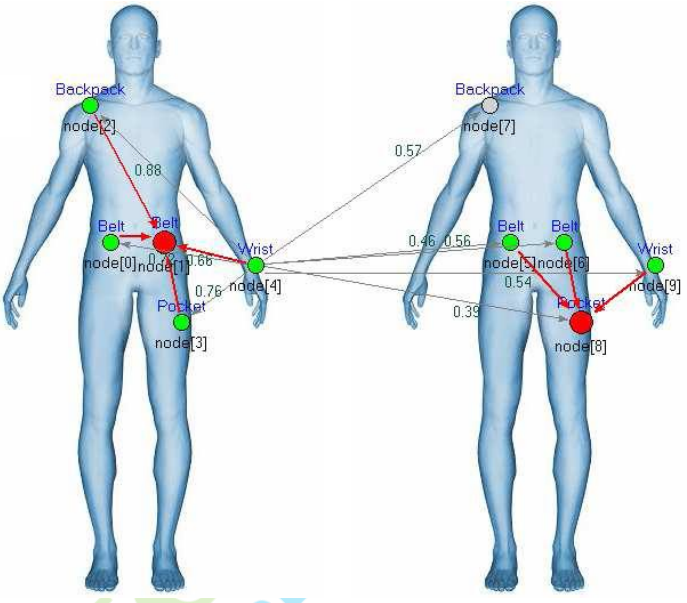


Fig. 2. Graphical simulation of the clustering algorithm on BANs

number. In any of these cases, v chooses as root node the neighbour root u with which it shares a common context and which has the highest priority number. If such a neighbour does not exist, v competes for clusterhead or becomes unassigned. The decision is based on the current status of the neighbours and tries to minimize the effect of the following erroneous situation: due to context fluctuations, an assigned node v may loose its root node and cannot join another cluster because none of its neighbours is root. Therefore, v may become root, form a new cluster and attract other nodes in that cluster. To avoid this undesirable outcome, a node declares itself root only if all its neighbours with which it shares a common context are unassigned. If there exists at least one neighbour u with which v shares a common context and u has a valid root node, then v becomes unassigned.

Node v announces the changes in choosing the root node by sending a local broadcast message *SetRoot* to its neighbours. In case of topological changes, this message is also used to announce the new neighbours of the current structure. The algorithm thus allows cluster merging or splitting, which meets the Requirement 1 from Section 4.1. We make the observation that there is no additional strain in terms of communication overhead on the clusterheads compared to the other nodes.

Let us consider the example from Figure 2. A BAN is associated with each person, consisting of five nodes placed in various locations: backpack, belt, pocket and wrist. The clustering structure is seen from the perspective of node 4, which is attached to the wrist of the left person. The clusterheads are represented with big circles (nodes 1 and 8). The dark-coloured arrows indicate the assignment

of the other nodes to the current clusterheads. Node 7 is unassigned, as the shared-context recognition algorithm did not associate this node with any of the neighbouring clusterheads at the previous time step. The light-coloured arrows show the confidence values computed by node 4 at the current time step in this particular case. The confidence values for the nodes on the same body with node 4 range between 0.66 and 0.88, while for the other body they lie between 0.39 and 0.57. Because the confidence of sharing the same context with the root node 1 is 0.66 and above the threshold, node 4 keeps the current root. Otherwise, it would have become unassigned (node 4 has some neighbours with the same context, having a valid root node), or assigned to the other cluster, if the confidence value for the neighbouring root node 8 was higher than the threshold.

5 Cluster Stability Analysis

Several algorithms for context sharing have been proposed in the literature, using various sensors and providing different accuracies (see Section 3). However, none of them gives a measure of the overall accuracy of the system, when multiple nodes sharing different contexts come together. We would like to analyse the cluster stability from both the theoretical point of view, by giving average approximation, upper and lower bounds, and through simulations. In addition, we are interested in the tradeoff between the time history necessary to achieve a certain stability and the responsiveness of the clustering algorithm. First, we compute the probabilities of correctly assessing the context, given the distribution of the confidence values. Second, we model the algorithm using Markov chains and we derive an approximation for the proportion of time the clustering structure is in a correct state.

5.1 Determination of Common Context

In this section, we give an example of how the probabilities of correct detection of the shared context can be computed.

Let v be a node in the network and u a neighbour of v. If v does not share the same context with u (e.g. they represent sensor nodes attached to different persons), we model the confidence value computed by the shared-context recognition algorithm with the random variable $X_1(v,u)$. If v shares the same context with u (e.g. they are attached to the same person), we model the confidence value as a random variable $X_2(v,u)$. We take the distribution encountered during the experiments as the reference Probability Density Function (PDF): we associate the random variables $X_1(v,u)$ with the PDF φ_1 and the corresponding Cumulative Distribution Function (CDF) Φ_1. Similarly, we associate the random variables $X_2(v,u)$ with the PDF φ_2 and CDF Φ_2.

Node v selects the subset of its neighbours with which it shares a common context based on a threshold value T. We choose T as the intersection point of the two PDFs φ_1 and φ_2, which minimizes the sum of probabilities of an incorrect determination. We denote p as the probability of a correct detection of a *common*

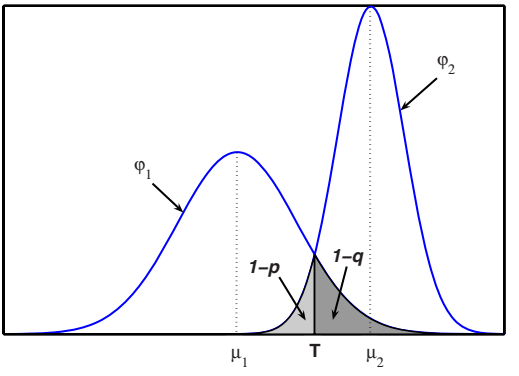


Fig. 3. The calculation of the threshold value T and the probabilities p and q

context and q as the probability of a correct detection of *different* contexts. The probabilities p and q are computed in the following way (see Figure 3):

$$p = 1 - \Phi_2(T), \qquad q = \Phi_1(T) \tag{1}$$

We compute the threshold value for the case where the distributions are normal, which is valid for the applications described in Section 2 (see the experimental data reported by [7,10]). Let us consider two normal distributions, $\varphi_1(\mu_1, \sigma_1)$ and $\varphi_2(\mu_2, \sigma_2)$. The intersection point of φ_1 and φ_2 which lies between μ_1 and μ_2 is the following:

$$T = \frac{\mu_1\sigma_2^2 - \mu_2\sigma_1^2 + \sigma_1\sigma_2\sqrt{(\mu_1 - \mu_2)^2 + 2(\sigma_2^2 - \sigma_1^2)ln(\sigma_2/\sigma_1)}}{\sigma_2^2 - \sigma_1^2} \tag{2}$$

Using Eq. 1 and 2, it is straightforward to compute p and q, knowing the characteristics of φ_1 and φ_2. We are now interested in how these probabilities change if we involve the time history in the decision process. The probability p_h of the correct detection that two nodes share a common context for a minimum time history h_{min} out of a total of H time steps is given by the CDF of the binomial distribution:

$$p_h(h_{min}, H) = \sum_{k=h_{min}}^{H} \binom{H}{k} p^k (1-p)^{H-k} \tag{3}$$

Similarly, the probability q_h of the correct detection of different contexts for a minimum time history h_{min} out of a total of H time steps is:

$$q_h(h_{min}, H) = \sum_{k=h_{min}}^{H} \binom{H}{k} q^k (1-q)^{H-k} \tag{4}$$

We have therefore $p = p_h(1, 1)$ and $q = q_h(1, 1)$.

5.2 Modelling with Markov Chains

We approximate the behaviour of the algorithm with a Markov chain, which allows us to estimate the global probability of having a correct cluster. We stress on the difference between a *time step* and a *Markov chain step*. A time step is related to the periodic update of the context information by the shared-context recognition algorithm which runs on every node. For improving the probabilities of a correct detection of a shared context, the algorithm looks over a time history H, composed of a number of time steps (see Section 5.1). A Markov chain step is the "memoryless" transition from one state to another, which happens on intervals equal to the total time history H.

We define a *group* G as the collection of nodes that share the same context in reality. We define a *cluster* C as the collection of nodes which have the same root node (as a result of Agorithm 1). The goal of the clustering algorithm is that for any group of nodes G, there exists a cluster with the root $r_0 \in G$ such that $\forall v \in V, r(v) = r_0 \Leftrightarrow v \in G$. Taking the example from Figure 2, we have two groups: $G_1 = \{0, 1, 2, 3, 4\}$, $G_2 = \{5, 6, 7, 8, 9\}$ and two clusters: $C_1 = \{0, 1, 2, 3, 4\}$ with root node 1, $C_2 = \{5, 6, 8, 9\}$ with root node 8.

We define the following states for a cluster C:

1. *Correct:* The cluster has exactly one root node from the group and all the members of the group are part of the cluster[1].
2. *Has root:* At least one group member is root.
3. *No root:* None of the group members is root.
4. *Election:* After reaching state 3, members of the group start an election process for choosing the root node.

For example, cluster C_1 from Figure 2 is *Correct*, while C_2 is in the state *Has root*, since node 7 is unassigned.

The transition matrix that determines the Markov chain is the following:

$$P = \begin{pmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \\ p_{41} & p_{42} & p_{43} & p_{44} \end{pmatrix}$$

Let $m \geq 0$ be the number of root nodes with higher priority than the current root and $k \geq 1$ the number of nodes in the cluster. If the cluster has a root, let r_0 be the root node. The probabilities p_{ij} are evaluated in a worst case scenario, by minimizing the chance to get in the *Correct* state.

The conditions which determine the probabilities p_{ij} are the following:

– p_{11}: (a) r_0 remains root in the next step, as r_0 does not share the same context with other root nodes with higher priority, and (b) all the nodes in the group share the same context with r_0.

[1] We intentionally take $G \subseteq C$. The nodes from $C \setminus G$ are part of other groups, which have the corresponding clusters in an incorrect state.

Table 1. Transition probabilities p_{ij}

Probability	Value	Probability	Value
p_{11}	$q^m p^{k-1}$	p_{21}	$q^m p^{k-1} q^{m(k-1)}$
p_{12}	$q^m(1-p^{k-1})$	p_{22}	$q^m(1-p^{k-1}q^{m(k-1)})$
p_{13}	$1-q^m$	p_{23}	$1-q^m$
p_{14}	0	p_{24}	0
p_{31}	0	p_{41}	0
p_{32}	0	p_{42}	$q^{mk}(1-(1-p)^{k(k-1)})$
p_{33}	0	p_{43}	0
p_{34}	1	p_{44}	$1-q^{mk}(1-(1-p)^{k(k-1)})$

- p_{12}: (a) r_0 remains root in the next step, and (b) there exists at least one node in the group that does not share the same context with r_0.
- p_{13}: r_0 shares the same context with a root node with higher priority, so that it gives up its role and joins another cluster.
- p_{21}: (a) r_0 remains root in the next step, (b) all the nodes in the group do not share the same context with other root nodes with higher priority and (c) all the nodes in the group share the same context with r_0.
- p_{23}: r_0 shares the same context with a root node with higher priority, so that it gives up its role and joins another cluster.
- p_{34}: from state *No root* the system goes at the next step to state *Election*.
- p_{42}: (a) all the nodes in the group do not share the same context with any root node with higher priority, and (b) there are at least two nodes in the group that share the same context.

Table 1 gives the computed probabilities for each transition of the Markov chain. We notice that the *Correct* state can be reached only from the *Has root* state. If $m > 0$, the probability p_{21} is minimized, so that in the stationary distribution of the Markov chain, the probability to be in the *Correct* state is lower than the real probability. Calculating the fixed row vector of the Markov chain yields the following result:

$$p_1(m,k,q,p) = \frac{p_{21}p_{42}}{(1+p_{21}-p_{11})(p_{42}+p_{42}p_{13}+p_{13})} \tag{5}$$

We define the *cluster stability* P_S as the probability of a cluster to be in the *Correct* state. Given that there are c clusters in the network, we have the following lower and upper bounds:

$$p_1(c-1,k,q,p) \leq P_S \leq p_1(0,k,q,p) \tag{6}$$

An estimation of the average case is given by:

$$P_S \approx p_1(\frac{c-1}{2},k,q,p) \tag{7}$$

6 Results

We analyse the performance of Tandem by running a series of simulations in the OMNeT++ [16] simulation environment. As the cluster formation and stability is affected only by the nodes in the one-hop neighbourhood, we simulate a network where the nodes are attached to different mobile objects or people, and where there is a wireless link between any two nodes. We focus on a mobile scenario with clustered nodes moving around and passing each other, and we analyse how the cluster stability changes when we vary the number of groups. First, we compare Tandem with a traditional clustering method and point out the factors that determine the improved behaviour of our algorithm. Second, based on the experimental results from previous work, we evaluate Tandem with respect to cluster stability and communication overhead.

6.1 Comparison to Traditional Clustering

Compared to traditional clustering methods, conceived to run only at the network layer, Tandem groups the nodes with similar semantic properties, namely the shared context. The uncertainty in determining the shared context and the resulting variation in time lead to a high instability of the clustering structure. Therefore, an algorithm that generates stable clusters distinguishes itself from generic algorithms designed for ad-hoc networks by paying attention especially on how to minimize the effect of the variation of confidence values.

To illustrate this difference, we present an alternative clustering algorithm following the idea of DMAC [2], denoted by DMAC*. In DMAC*, each node v takes into account every change in the confidence value. DMAC* has the following two main differences compared to Tandem:

1. If v detects in its neighbourhood a clusterhead r_1 with a higher priority value than the current clusterhead r_0, and v shares a common context with r_1, then v directly chooses r_1 as clusterhead.
2. If none of its neighbours sharing a common context is clusterhead, then v declares itself a clusterhead.

The first condition may lead to unwanted effects, where a single error in the detection of different contexts between v and r_1 changes the cluster membership of v. Likewise, an error in the detection of the common context between v and r_0 results through the second condition into the erroneous situation of having two clusterheads for the same group.

We show in Figure 4 a comparison between Tandem and DMAC* from the stability and energy efficiency points of view. Each group is composed of 10 nodes and we vary the number of groups between 2 and 10. We take $p = q = 0.99$. The energy spent by Tandem on communication is proportional to the number of *SetRoot* messages sent by every node in one time step. Compared with the results obtained for DMAC*, Tandem improves the stability up to 18% on an absolute scale and the energy efficiency up to 32% on a relative scale, thus satisfying Requirements 2 and 3 from Section 4.1.

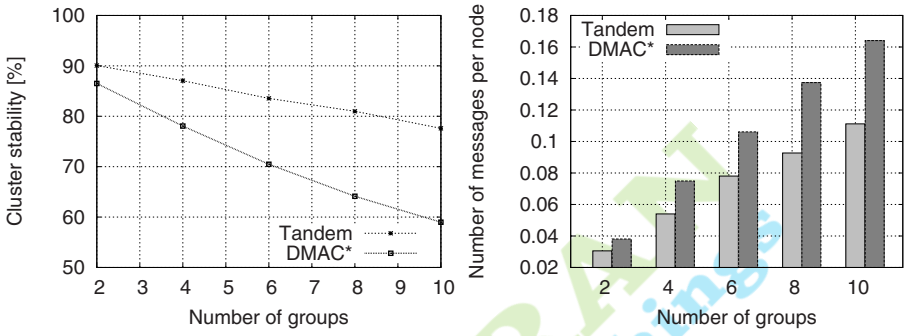


Fig. 4. Comparison between Tandem and a clustering alternative

6.2 Evaluation of Tandem

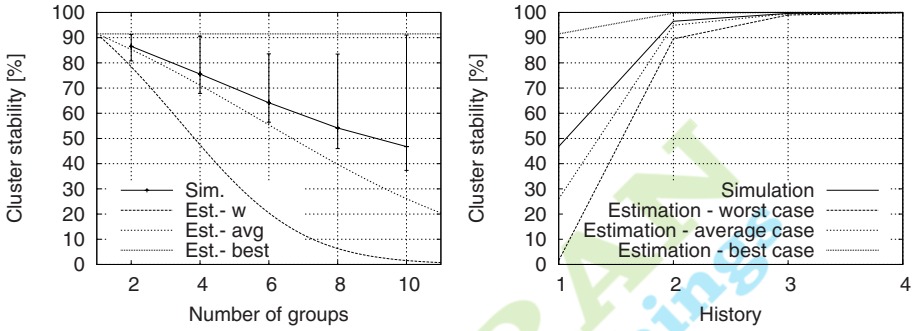
A typical example of context sharing is the similarity of movement, which we analyse in this section using experimental results corresponding to the scenarios described in Section 2. In general, the movement information is extracted from accelerometers. Simpler sensors such as tilt switches can be also used, but with less accurate results. We have the following two concrete examples of wireless objects moving together:

1. **RTI** - wireless sensor nodes used in a transport scenario, which correlate their movement pattern; the shared-context recognition algorithm computes a correlation coefficient between the movement data of two neighbouring nodes, which is extracted using both tilt switches and accelerometers [10].
2. **BAN** - smart devices that decide whether they are carried by the same person; the shared-context recognition algorithm uses a coherence function of the movement data provided by accelerometers [7].

Table 2 shows the characteristics of the normal distributions derived from the concrete experiments conducted in both application examples, together with the computed threshold from Eq. 2 and the probabilities p and q. Contrary to the RTI scenario, where the nodes moving together experience exactly the same movement pattern, in the BAN scenario different parts of the body are engaged in different types of movements during walking. For a realistic evaluation, we choose the worse case experimental results from the BAN scenario, where the nodes are attached to the pocket and wrist of the subjects (Pocket/Wrist trial [7]).

Table 2. Statistics from the experiments and computed probabilities

Application	μ_1	σ_1	μ_2	σ_2	T	p	q
RTI - tilt switch	0.641	0.087	-0.017	0.249	0.438	0.9902	0.9662
RTI - accelerometer	0.817	0.106	0.009	0.124	0.442	0.9998	0.9998
BAN - Pocket/Wrist	0.757	0.065	0.519	0.069	0.640	0.9636	0.9607



(a) $h_{min} = H = 1$ (b) $h_{min} = 1..4$, 10 groups

Fig. 5. Cluster stability in the RTI with tilt switches scenario

The scalability analysis of the movement correlation method proposed for the RTI scenario indicate a maximum network density of 100 nodes [10], imposed by the shared wireless medium. Because the same periodic transmission of the movement data is needed for the BAN scenario [7], we use in our simulations the same maximum network density for both applications. We have 10 nodes in each group moving together and interacting with other groups, and we vary the number of groups between 2 and 10 and also the time history. We recall from Section 5.2 that the cluster stability is the probability that the cluster is in the *Correct* state. We represent the cluster stability in percentage, for the following cases: (1) average simulation results, (2) estimation of the worst case, derived from Eq. 6, (3) estimation of the average case, derived from Eq. 7, and (4) estimation of the best case, derived from Eq. 6.

For each point on the simulation plots we run up to 10 simulations of $10^4 - 10^5$ time steps. In order to study the influence of the history size, we take $H = 2h_{min} - 1$ and we vary h_{min} from 1 to 4.

RTI with tilt switches scenario. Figure 5(a) shows the cluster stability depending on the number of groups present in the network, given that the time history is $h_{min} = 1$. The error bars represent the absolute minimum and maximum stability recorded during the simulations. We notice that the results respect the upper and lower bounds calculated theoretically. The estimation of the average case is close to the simulations for a small number of groups. However, increasing the number of groups decreases the precision of the approximation, due to the minimization of the transition probabilities to get in the *Correct* state (see Section 5.2).

Figure 5(b) shows the cluster stability depending on the time history, for a network composed of 10 groups. We notice that increasing the time history considerably improves the cluster stability and the theoretical approximation. For a time history $h_{min} = 4$ ($H = 7$), a stability of 99.92 is achieved, while the lower bound is 99.89. Therefore, for achieving a stability close to 100%, the necessary delay is $H \times 16 = 112$ seconds (the size of the data sequence is 16 seconds [10]).

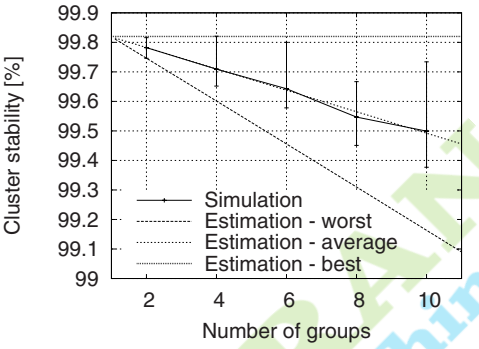


Fig. 6. Cluster stability in the RTI with accelerometers scenario ($h_{min} = H = 1$)

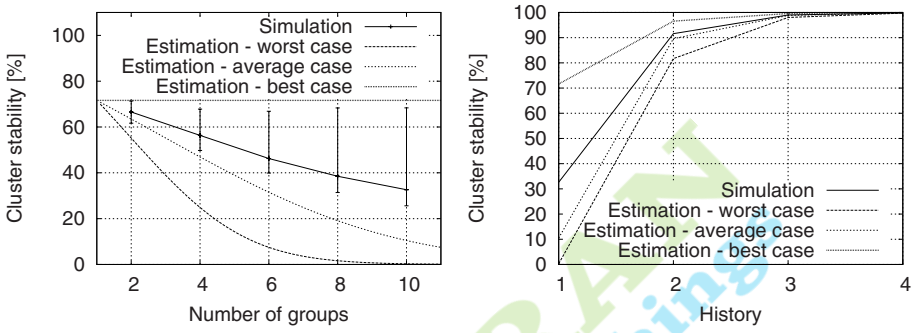
RTI with accelerometers scenario. The solution using accelerometers is more reliable, resulting in higher probabilities for a correct detection of the context (see Table 2) and consequently, higher cluster stability. Figure 6 shows the cluster stability depending on the number of groups present in the network, given that the time history is $h_{min} = 1$. We also represent the error bars for the absolute minimum and maximum values. We notice the high stability obtained even for large number of groups (99.5 for 10 groups). Due to the fact that the clusters stay in the *Correct* state for most of the time, the approximations are close to the simulation results. For this scenario, a high cluster stability can be achieved even considering the time history 1, reaching a delay of only 16 seconds.

BAN scenario. Figure 7(a) shows the cluster stability in the BAN scenario, depending on the number of groups, given that the time history is $h_{min} = 1$. Similarly with the two scenarios presented above, we notice that the results respect the upper and lower bounds calculated theoretically. The average stability is lower than in the previous cases, with a maximum of 67% and less than 50% for a network composed of more than 6 groups.

Figure 7(b) shows the cluster stability depending on the time history, for a network composed of 10 groups. The time history significantly improves the cluster stability: for the time history $h_{min} = 4$ ($H = 7$), the stability is 99.84 and the lower bound is 99.74. For achieving this, the delay is $H \times 8 = 56$ seconds (the window size is 8 seconds [7]).

Communication overhead. The communication overhead is induced by the *SetRoot* message, sent by every node when the current root node changes. In case the lower networking layers use a periodic message in order to maintain for example the synchronization (e.g. in case of a TDMA MAC protocol), Tandem can use this heartbeat to piggyback the small piece of information present in the *SetRoot* message (root address and priority number). Assuming that there is no such periodic message, we count the average number of *SetRoot* messages sent by each node in one step of simulation on average.

Figure 8(a) shows the number of messages on a logarithmic scale, depending on the number of groups. The results correspond to each of the three scenarios,



(a) $h_{min} = H = 1$ (b) $h_{min} = 1..4$, 10 groups

Fig. 7. Cluster stability in the BAN scenario

where the time history is 1. We notice that the more stable the structure is, the less communication overhead is needed. For the RTI with accelerometers scenario, less than 1 message is sent per node in 100 time steps, even for a large number of groups. The overhead is increasing as the number of groups increases, due to the diminishing cluster stability.

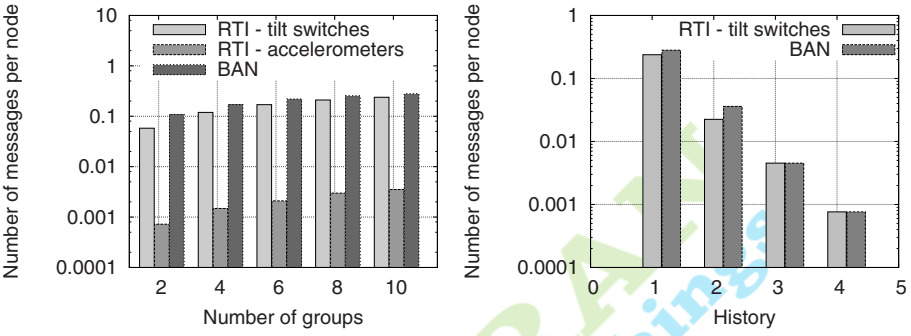
Figure 8(b) shows the number of messages depending on the time history, for the RTI with tilt switches and BAN scenarios. Increasing the time history improves the stability and thus reduces the communication overhead. For the time history 4, the overhead is less than 10^{-3} messages per node.

7 Discussion and Conclusions

We presented Tandem, a context-aware method for spontaneous clustering of wireless sensor nodes. The algorithm allows reclustering in case of topological or contextual changes. By analysing the similarity of the context over a time history, Tandem tries to achieve stable clusters. We approximate the behaviour of the algorithm using a Markov chain model and we analyse the cluster stability theoretically and through simulations, using experimental results reported from real field tests. The analysis gives the possibility to theoretically estimate the stability of the structure and the responsiveness of the algorithm. Computing the worse case stability via the Markov chain approximation, we can deduce the time history necessary to achieve stable clusters. In what follows, we discuss the main advantages and limitations of the proposed clustering method.

Advantages

- *Responsiveness.* The clustering structure reacts quickly to topological and contextual changes: nodes decide based only on the current situation of their neighbourhood, without the need of any negotiation with other parties.
- *Small-scale experiment required.* For computing the probabilities p and q that can be used to estimate the cluster stability, only a small-scale reproducible



(a) Number of groups (b) History

Fig. 8. Average number of *SetRoot* messages sent per node in 100 steps

experiment is required. For example, two nodes moving together and another two moving separately are enough to generate the statistical distributions of the confidence values.

– *Delay estimation.* By deducing the time history required to achieve a certain stability, the delay in accommodating the topological or contextual changes can be easily estimated.

Limitations

– *Rough approximation for many groups.* As we notice from Figures 5(a) and 7(a), the difference between the approximation that we derive using Markov chains and the real situation is increasing with the number of groups. However, the model offers a good approximation in case of highly accurate context detection methods (see Figure 6). Therefore, the approximation can be successfully used for deducing the minimum time history for a cluster stability close to 100%.

– *Multihop clusters.* The method that we propose is valid only for one-hop clusters, which is justified taking into account the scenarios from Section 2. Nevertheless, other applications may require multihop clusters, even several layers of clustering. For example, groups of people skiing together, forming multihop clusters, where each person is wearing a BAN that is a one-hop cluster. The algorithm can be easily extended to accommodate multihop clusters: instead of choosing directly the clusterhead node, every node selects a parent and thus joins the cluster associated with the parent node.

For future work, we intend to extend the algorithm for multihop clusters and to investigate the cluster stability in this case. We also plan a series of experiments involving clustering of nodes on body area networks. Subsequent to clustering, a task allocation mechanism distributes various tasks to the nodes which are part of the cluster, depending on their capabilities. The final goal

is to have a distributed activity recognition algorithm running on a dynamic, context-aware BAN.

References

1. Antifakos, S., Schiele, B., Holmquist, L.E.: Grouping mechanisms for smart objects based on implicit interaction and context proximity. In: Dey, A.K., Schmidt, A., McCarthy, J.F. (eds.) UbiComp 2003. LNCS, vol. 2864, pp. 207–208. Springer, Heidelberg (2003)
2. Basagni, S.: Distributed clustering for ad hoc networks. In: ISPAN 1999, pp. 310–315. IEEE Computer Society, Los Alamitos (1999)
3. Bouhafs, F., Merabti, M., Mokhtar, H.: A semantic clustering routing protocol for wireless sensor networks. In: Consumer Communications and Networking Conference, pp. 351–355. IEEE Computer Society, Los Alamitos (2006)
4. Gluhak, A., Presser, M., Zhu, L., Esfandiyari, S., Kupschick, S.: Towards mood based mobile services and applications. In: Kortuem, G., Finney, J., Lea, R., Sundramoorthy, V. (eds.) EuroSSC 2007. LNCS, vol. 4793, pp. 159–174. Springer, Heidelberg (2007)
5. Gu, L., Jia, D., Vicaire, P., Yan, T., Luo, L., Tirumala, A., Cao, Q., He, T., Stankovic, J.A., Abdelzaher, T., Krogh, B.H.: Lightweight detection and classification for wireless sensor networks in realistic environments. In: SenSys, pp. 205–217. ACM Press, New York (2005)
6. Holmquist, L.E., Mattern, F., Schiele, B., Alahuhta, P., Beigl, M., Gellersen, H.-W.: Smart-its friends: A technique for users to easily establish connections between smart artefacts. In: Abowd, G.D., Brumitt, B., Shafer, S. (eds.) UbiComp 2001. LNCS, vol. 2201, pp. 116–122. Springer, Heidelberg (2001)
7. Lester, J., Hannaford, B., Borriello, G.: Are You with Me? - using accelerometers to determine if two devices are carried by the same person. In: Ferscha, A., Mattern, F. (eds.) PERVASIVE 2004. LNCS, vol. 3001, pp. 33–50. Springer, Heidelberg (2004)
8. Lorincz, K., Malan, D.J., Fulford-Jones, T.R.F., Nawoj, A., Clavel, A., Shnayder, V., Mainland, G., Welsh, M., Moulton, S.: Sensor networks for emergency response: Challenges and opportunities. IEEE Pervasive Computing 03(4), 16–23 (2004)
9. Marin-Perianu, M., Havinga, P.J.M.: D-FLER: A distributed fuzzy logic engine for rule-based wireless sensor networks. In: Ichikawa, H., Cho, W.-D., Satoh, I., Youn, H.Y. (eds.) UCS 2007. LNCS, vol. 4836, pp. 86–101. Springer, Heidelberg (2007)
10. Marin-Perianu, R.S., Marin-Perianu, M., Havinga, P.J.M., Scholten, J.: Movement-based group awareness with wireless sensor networks. In: LaMarca, A., Langheinrich, M., Truong, K.N. (eds.) Pervasive 2007. LNCS, vol. 4480, pp. 298–315. Springer, Heidelberg (2007)
11. Mayrhofer, R., Gellersen, H.: Shake well before use: Authentication based on accelerometer data. In: LaMarca, A., Langheinrich, M., Truong, K.N. (eds.) Pervasive 2007. LNCS, vol. 4480, pp. 144–161. Springer, Heidelberg (2007)
12. Siegemund, F.: A context-aware communication platform for smart objects. In: Ferscha, A., Mattern, F. (eds.) PERVASIVE 2004. LNCS, vol. 3001, pp. 69–86. Springer, Heidelberg (2004)
13. Stiefmeier, T., Lombriser, C., Roggen, D., Junker, H., Ogris, G., Tröster, G.: Event-based activity tracking in work environments. In: Proceedings of the 3rd International Forum on Applied Wearable Computing (IFAWC) (March 2006)

14. Stojmenovic, I., Seddigh, M., Zunic, J.: Dominating sets and neighbor elimination-based broadcasting algorithms in wireless networks. IEEE Transactions on Parallel and Distributed Systems 13(1), 14–25 (2002)
15. Strohbach, M., Gellersen, H.: Smart clustering - networking smart objects based on their physical relationships. In: Proceedings of the 5th IEEE International Workshop on Networked Appliances, pp. 151–155. IEEE Computer Society, Los Alamitos (2002)
16. Varga, A.: The omnet++ discrete event simulation system. In: ESM 2001, Prague, Czech Republic (June 2001)
17. Wang, J., Chen, G., Kotz, D.: A sensor fusion approach for meeting detection. In: MobiSys 2004 Workshop on Context Awareness (June 2004)

IOT IRAN Internet of Things

Stream Feeds - An Abstraction for the World Wide Sensor Web

Robert Dickerson, Jiakang Lu, Jian Lu, and Kamin Whitehouse

Department of Computer Science
University of Virginia
rfdickerson,jklu,jl3aq,whitehouse@cs.virginia.edu

Abstract. RFIDs, cell phones, and sensor nodes produce streams of sensor data that help computers monitor, react to, and affect the changing status of the physical world. Our goal in this paper is to allow these data streams to be first-class citizens on the World Wide Web. We present a new Web primitive called *stream feeds* that extend traditional XML feeds such as blogs and Podcasts to accommodate the large size, high frequency, and real-time nature of sensor streams. We demonstrate that our extensions improve the scalability and efficiency over the traditional model for Web feeds such as blogs and Podcasts, particularly when feeds are being used for in-network data fusion.

1 Introduction

The *Internet of Things* (IOT) refers to the vision in which computers on the Internet are able to monitor, react to, and affect the changing status of objects in the physical world. The two major forces driving the IOT are (i) the widespread adoption of RFIDs that allow physical objects to be uniquely identified, and (ii) rapid proliferation of portable devices like cellphones and wireless sensor nodes that push computation, communication, and sensing deeper into the physical world. Both of these new technologies produce streams of data that must be stored and processed.

In this paper, we address the issue of connecting these data streams with the World Wide Web. Although the terms *Web* and *Internet* are often used interchangeably, they are quite distinct: the Internet is a set of computers connected by the IP protocol, while the Web is a set of data resources that are connected by hyperlinks. Our goal in this paper is to allow data streams to be first-class citizens on the Web: every data stream has a URL, is manipulated through the HTTP protocol, and is hyperlinked to other objects on the Web. This would allow data streams to easily interoperate with other web applications and enable crawling and indexing by search engines. In addition, streams will inherit other Web features such as authentication and encryption.

There are currently several abstractions for stream-like objects on the Web. For example, a Web *feed* is an XML document that contains a dynamically changing sequence of content items, such as blog entries or news headlines. Feeds are transported as a single file using the *pull* mechanism provided by HTTP

C. Floerkemeier et al. (Eds.): IOT 2008, LNCS 4952, pp. 360–375, 2008.
© Springer-Verlag Berlin Heidelberg 2008

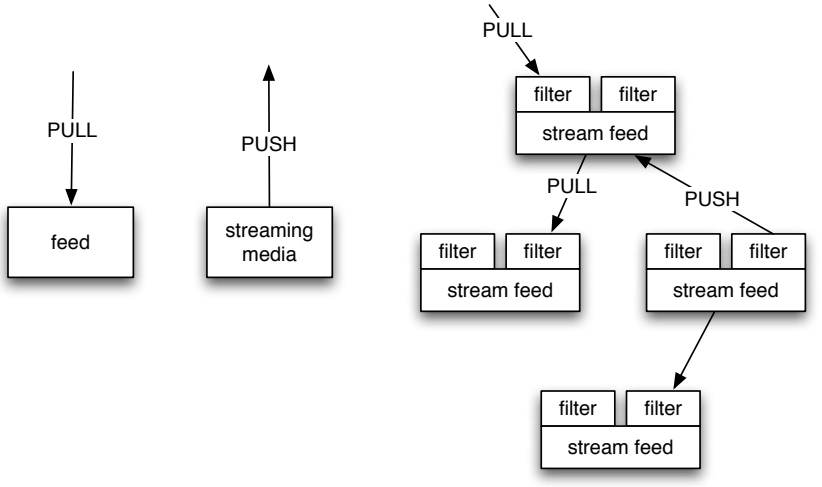


Fig. 1. Web feeds (left) only support a PULL abstraction while streaming media (middle) only supports a PUSH abstraction. The stream feed abstraction (right) must support both types of data transport, must provide server-side filtering, and must support data fusion trees.

requests. On the other hand, multimedia streams such as Internet radio stations are transported in real-time as the content is generated, using a *push* mechanism provided by protocols like the Real Time Streaming Protocol (RTSP). This protocol allows operations like "play" and "pause" on streaming media objects.

The problem we address in this paper is that sensor streams do not fit into any of the stream abstractions currently available on the Web. Clients will sometimes want to *pull* the historical data of a sensor stream, similar to a Web feed, but other clients will want real-time updates, similar to a media stream. Furthermore, many clients will only want a small part of the total stream. Finally, unlike Web feeds or media feeds, sensor streams will commonly be *fused* to create new sensor streams.

We hypothesize that we could provide an abstraction that meets the special needs of sensor streams by adding a few simple extensions to Web feeds. To test this hypothesis, we implement a new Web abstraction called the *stream feed*. Just like normal feeds, stream feeds can be accessed through a URL and are accessed through a Web server over HTTP. As such, they can be both the source and the target of hyperlinks. They are dynamic objects that are retrieved using non-streaming transport, so updates to the content can only be observed by retrieving the entire object again. However, stream feeds also support two extensions not supported by normal feeds: server-side filtering and streaming transport. These extensions allow stream feeds to be very large objects that change with very high frequencies and are served to clients with real-time updates.

Stream feeds can be *fused*, processed and filtered to create new stream feeds, and this process can be repeated to create a *fusion tree*. For example, data from

multiple sensors on a highway can be combined to create new streams that indicate vehicle detections on the road, and these streams can in turn be fused to create a new stream with traffic conditions over time. This is different from the *aggregation* of other feeds in which, for example, news headlines from multiple sources are combined into a single feed. We demonstrate that the traditional model of serving feeds does not scale well with the number of clients and the size of the feeds, particularly with deep fusion trees, and that our stream feed extensions improve scalability, latency, and the total amount of bandwidth consumed. These results support the conclusion that stream feeds can serve as the basis for a World Wide Sensor Web in which sensor data from around the world can be shared and processed within a single global framework.

2 A Motivating Application

MetroNet is a sensor system that we are currently designing to measure foot traffic near store fronts using commodity sensors such as motion sensors and thermal cameras in downtown Charlottesville where shops are densely packed and there is high pedestrian traffic at points in the day. This simple system will be used by shops to identify the effectiveness of window displays, signs, or other techniques designed to attract foot traffic into the shop. It will measure (i) how many people walk by the shop (ii) how many people enter the shop. The ratio of these values is extremely valuable to retail stores. In addition, many stores consider the *conversion rate*, the store's number of retail transactions during a certain time period compared to its traffic, as good indicator of its performance. Measuring foot traffic can be helpful when evaluating proper labor usage, advertising and promotional programs, prime selling hours, visual merchandising strategies, future opportunities or challenges, new store concepts, store location analysis, and merchandise assortment and stock levels.

The MetroNet sensors provide valuable data to the shopkeepers, but there are also a number of compelling reasons to *share* the data. For example, shopkeepers could use data from neighboring shops to normalize their own data, allowing them to differentiate the effect of a bad storefront display from the effect of bad weather, vehicular traffic, and downtown events such as fairs or concerts. Once in place, the sensors and their data would also be extremely valuable to the residents and planners in the city. For example, potential residents that would like to buy a new house could query for aggregate activity levels in each part of the city to find a location that is close to the city's night life. This type of query could be augmented in a subsequent phase by collecting supplemental data such as average levels of noise, sunlight, humidity, or air pollutants. Potential business owners would have empirical data upon which to base the value of commercial real estate, and city planners could gear zoning and traffic decisions to increase business and quality of life for residents. In fact, one motivation for the MetroNet testbed is to provide empirical data for an ongoing debate in Charlottesville about the effect on business of vehicular traffic crossing the downtown pedestrian zone.

IoT IRAN Internet of Things

Because MetroNet data can be useful to many parties, it could be used to form *fusion trees*. For example, the shopkeepers might publish streams of raw sensor data, but this sensor data may be *fused* by third parties to produce the counts of people walking in front of and into the shops. These third parties may then publish this information on their own servers as streams of people detection events. These new streams may then be fused again to produce streams with commercial real estate values or real-time estimates of overall pedestrian activity levels. A client query at the top of a tree would cause a sequence of cascading queries that consume network bandwidth and increase the latency of the query response. Stream feeds must be able to support queries over fusions trees quickly and efficiently.

There are many different requirements under which MetroNet data and the data streams that are created from it could be shared. Some data consumers, such as city planners who are moving a road to minimize effects on pedestrian traffic, may want the sensor streams as a bundle of historical data. Sometimes, the sensor stream objects could be very large (on the order of Terabytes), and so the city planners would like to query for only a small part of the entire stream, such as the data collected in the past year. Other data consumers, such as those looking at a web page with current estimates of pedestrian activity, may require real-time updates to the sensor streams. Furthermore, some data streams could be completely private, shared with other shopkeepers, or shared only with storefronts that are owned by the same company. This type of sharing may require authentication and encryption. Data could also be shared publicly, in which case the data consumer must be able to find the data through some sort of index, such as a search engine.

This case study demonstrates that stream feeds must accommodate several needs:

- *Historical queries*: many applications demand analysis over past data, and should be accessible if a server has stored it.
- *Real-time updates*: some events have low frequency but real-time importance upon firing.
- *Server-side filtering*: only a subset of data should flow from source to destinations to lessen network load.
- *Authentication*: some aggregates should be made from authorized users
- *Encryption*: some branches of the fusion tree should remain private to a certain group.
- *Indexability*: keywords from the stream meta data should be indexable along with the URL of the stream itself.
- *Scalability*: The system performance should not deteriorate dramatically as the number of clients increases, or as the depth of a fusion tree increases.
- *Autonomy*: a server should not be required to store its data on a third-party's server, and vice versa.

3 Background and Related Work

We can define a *sensor stream* to be a set of data that is generated over time by a single sensor. For example, a stream could be the set of temperature data generated by a temperature sensor over time. A stream is distinct from a *connection*, which is a software abstraction through which one device may send data to another at any time. For example, a TCP socket is a connection, but is not a stream. *Multimedia streaming* is a common technique in which a media object is transported such that it can be played before the entire file is downloaded. This is a distinct concept from a sensor stream: one property of steam feeds is that sensor streams can be transported using either streaming or non-streaming connections.

Another important distinction for this paper is the difference between the World Wide Web and the Internet. The *Web* is a set of objects that are uniquely addressed by a uniform resource locator (URL), are hosted by a *web server*, and are accessed via the HTTP protocol. Objects on the web refer to each other by their URL's, creating *links* that form a directional graph. Links in HTML documents are called *hyperlinks* while links in XML documents are called *XLinks*. The directional graph formed by these links help users to locate objects on the Web in a process called *surfing*, allow automated bots to *crawl* the Web, and form the basis for indexing and search algorithms such as Google's PageRank algorithm [1]. Although the Web is often also called the Internet, it is a distinct concept: the *Internet* is a set of computers connected by the IP protocol, while the Web is a set of objects connected by links. Most Web objects can be accessed through the Internet, but not all Internet applications are on the Web. As we describe below, most architectures that have been proposed for the World Wide Sensor Web exploit the globally-connected nature of the Internet, but do not actually put sensor streams on the Web: data streams do not have a URL, are not served by Web servers over HTTP, and cannot be the subject or object of a link. The main contribution of the stream feeds abstraction is that high data rate, aperiodic, real-time data streams become first-class citizens on the Web.

3.1 XML Schemas for Sensor Streams

The problem of sharing sensor data streams is currently becoming an active area of research. The Open Geographic Consortium (OGC) is an organization of 352 companies, government agencies, and universities collaborating to design publicly available interface specifications. The SensorNet [2] project primarily uses the OGC's schemas to achieve web interoperability. One example schema is the SensorML specification, which is an XML schema used to define processes and components associated with measurement of observations. These schemas are being created to define the a standard model for Web-based sensor networks, which will enable the discovery, exchange, processing of sensor observations, and tasking of sensor systems. However, they are not associated with a proposed architecture to store, cache, filter, and fuse data streams, and to translate between the real-time streaming nature of data streams and the static, connectionless nature of the Web.

Other data formats include GeoRSS, which is used for encoding geographic objects in RSS fees and KML (Keyhold Markup Language), a similiar format especially designed for GoogleEarth and Google Maps [3]. However, these groups are only proposing data formats, they are not proposing an architecture or abstraction that will integrate these XML formats with the Web. Stream feeds are agnostic with respect to XML Schemas for sensor streams and any XML format may be used. Some architectures have been designed to integrate SensorML files with the Web, for example, in the Davis Weather System, AIRDAS airborne sensor, and the SPOT satellite. However, these implementations do not propose solutions to the criteria we laid out in Section 2, since they do not define complex interrelationships between data sources that are owned by several entities.

3.2 Web Feeds

A Web *feed* is an XML document that contains a dynamically changing sequence of content items, such as blog entries, Podcasts, or news headlines. Content providers *syndicate* their web feeds and content consumers *subscribe* to that feed. A subscription is usually done through a feed *aggregator*, which continually queries for the feed at regular intervals and processes it or notifies the user when the feed content has been updated. Using this model, there is no coupling between the publisher and the consumer: subscriptions can be broken by removing the feed from the user's aggregator. This model has worked well for feeds until now because traditional feeds are relatively small data objects and have little or not realtime requirements. This small size also makes it possible to filter the feed content within the client aggregator, instead of on the server side. Stream feeds differ from traditional feeds because they may often be much larger data objects, they may change with orders of magnitude higher frequency than traditional feeds, and clients that request stream feeds may require immediate update. This combination makes periodic requests for the entire data object inefficient. Another difference is that the set of operations that is typically performed on the data content of traditional feeds is limited to concatenation, sorting, and truncation of content entries. The content of sensor streams, however, can be processed, filtered, and *fused* with content from other sensor streams. There are systems that use XML files available on the web for disseminating sensor data both realtime and historical, resulting in a virtual sensor abstraction that provides a simple and uniform accesss to heterogeneous streams [**?**]. However, these systems do not provide a common interface for pushing and pulling for the data.

Research is interested in finding ways to search for data coming from these feeds. One example is Cobra which crawls, filters, and aggregates RSS feeds [5], and another is the popular search engine Technorati, designed to search for keywords in web feeds.

3.3 Streaming Media

The HTTP protocol was designed to retrieve an entire document at once, using a *pull* model for file transport. This model causes problems for large multimedia

files that can take a very long time to download. To address this problem, several protocols and techniques have been designed to *stream* media files in order to allow the file to be played by the client before the file has been completely downloaded. For example, some systems stream data through multiple sequential HTTP responses while others use new protocols like RTP (realtime protocol) and RTSP (realtime streaming protocol) [6]. Protocols like Jabber (XMPP) [7] support streams of instant messages and can be used to stream realtime data past firewalls. The WHATWG is drafting a Web Applications 1.0 specification [8] which server-side pushing of content to the client.

Streaming file transport is used for finite-sized media files, but it also enabled the creation of *streaming media*, a type of data object that has no end, like an Internet radio station. This type of data object cannot be used with the traditional *pull* model for file transport. Sensor streams are similar to streaming media in this respect. However, the streaming transport protocols that have been designed until do not meet all of the requirements for stream feeds. They support the abstraction of a streaming media object, and provide functions such as 'play' and 'pause'. Sensor streams do not fit under this abstraction, and require other commands such as the downloading of historical data and server-side data filtering.

3.4 RFIDs

RFID tags can uniquely identify a physical object as it moves through a supply chain that may involve dozens of different vendors and shippers. For this reason, several architectures have been designed to make RFID data available from multiple administrative domains. For example, EPCglobal has created a global system for managing and sharing RFID and sensor data [9]. This system uses an asynchronous messaging protocol based on Web service standards to make Electronic Product Codes (EPC) data available in real time to hundreds of thousands of simultaneous users. However, this work differs from ours in that we are creating an abstraction based on URLs, so sensor data can be linked to by other Web objects, and can be crawled and index by search engines.

RFID applications have a *high fan-in* architecture where there are many devices at the edge of the network sending their data to a small number of more powerful processing nodes. High fan-in systems can suffer from congestion, and a system called HiFi [10] addresses this problem by performing data cleaning, event monitoring, stream correlation, and outlier detection among the mid-tier nodes.

3.5 Other World Wide Sensor Web Architectures

There have been several designs for an infrastructure for global sensor data sharing. Most architectures propose either putting all sensor data into one large centralized database or distributing data across a distributed database running the same software stack. Centralized databases have advantages that come with simplicity, but centralization can lead to high server loads and require others to trust the server with private information. Centralization runs tangent to the

web's laissez-faire philosophy. The problem with database-oriented solutions is that they force the data producer to conform to a pre-specified implementation, and do not allow for diversity of software stacks. They also do not provide rank-ordered query results so they cannot deal with issues of data fidelity and popularity.

IrisNet is a distributed database that has been proposed as an architecture for the World Wide Sensor Web [11]. Sensor data is stored in an XML database which is hierarchically partitioned and stored in different parts of the network to minimize response time and network traffic during queries. IrisNet restricts data positioning to a sensor ontology based on a hierarchy of locations. Many other systems have also been designed for distributed stream processing using distributed hash tables[12] or by moving the data to a centralized location and running continuous queries [13,14,15].

The SensorMap project from Microsoft Research uses a database-oriented approach where users post queries to a central database by interacting with a graphical map interface [16]. SensorMap is similar to our project in that the thrust is to make publishing data especially easy. SensorMap has explored using *screen scraping* by enabling a web crawler to automatically index adjacent text descriptions next to embedded webcam content on a webpage.

Hourglass is a system that provides circuit-based connections for real-time persistent streams [17]. It emphasizes intermittent connection scenarios for mobile or poorly connected nodes. Circuits push data through a series of links and can be processed along the way. However, because all data is pushed, Hourglass only supports real-time data streaming and does not allow searching through the history of a data stream. In addition, Hourglass does not support indexing streams and all data queries must be made through their registry/query service. Hourglass does not require a particular data schema, like database systems, but it does require users to use a particular software stack.

4 The Stream Feed Abstraction

Stream feeds provide the abstraction of a dynamic content object containing a stream of sensor data. This data object can be located with a URL, can be linked to from other data objects, and can be accessed over the HTTP protocol. For example, to access a stream of temperature data from the front door of the Whitehouse, one would visit:

```
http://www.whitehouse.gov/streams/frontdoor/temperature
```

When the client requests this URL, the client opens a TCP socket to the server and sends a HTTP GET request, as usual. The server responds immediately with all data that is already contained in the sensor stream, followed by updates about the sensor stream in real-time as new data objects are added to the stream.

A parameter can be appended to this URI to indicate that the query should take place over a different time region. For example, the client can request:

```
/streams/frontdoor/temperature/historical
```

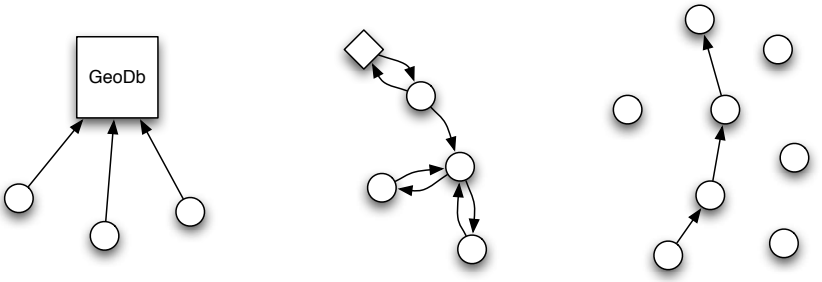


Fig. 2. SensorMap (left) collects all data into a central data base. IrisNet (center) provides a distributed XML database abstraction that can be queried by location. Hourglass (right) allows the user to setup *circuits* along which sensor data is pushed.

to restrict the query only to data that has already been stored, or can request

```
/streams/frontdoor/temperature/incoming
```

to restrict the query only to data that has not yet been received. Thus, the client can pose the query over three temporal scopes: (i) all time (default), (ii) all time before now, and (iii) all time after now. These three scopes are illustrated in Figure 3.

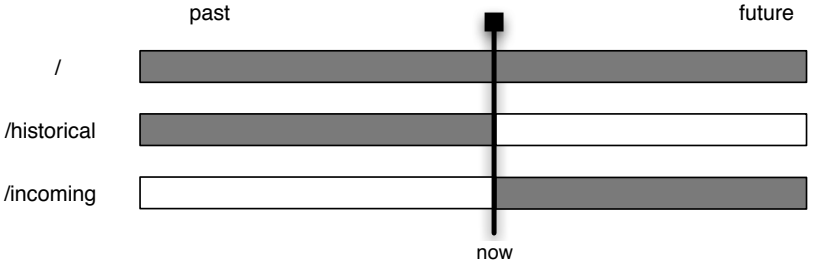


Fig. 3. Every query over a stream feed must take one of three temporal scopes. The default (top) is a query over all time, but queries over historical data (center) and incoming data (bottom) are also possible. Historical data is returned immediately while incoming data is returned as it arrives.

The user can also *filter* the sensor stream by passing filter parameters as follows:

```
/temperature/filter/?value_lower_bound=100&day_equal=Tuesday
```

This query restricts the query response to only contain the values in the data stream that have a "value" attribute greater than 100 and a "day" attribute equal to 'Tuesday'. The values that can be filtered depend on the XML schema that the sensor stream is using, so the client must be familiar with this schema. The values in the sensor stream are then filtered using syntactic filtering of the

data points based on the XML tag names and values. This can be performed automatically for any XML schema. filters can be applied to a query with any of the three temporal scopes shown in Figure 3. Any tag can be used with four types of filters:

```
TAG_equal
TAG_not_equal
TAG_lower_bound
TAG_upper_bound
```

The URL-based stream feed interface conforms to REST (Representational State Transfer) principles describing how resources are defined and addressed. Unlike RPC-based web services, where a vast set of verbs to be applied to a set of nouns, the RESTful philosophy promotes the use of a simple set of verbs that operate on a vast collection of nouns. The advantage to using a RESTful interface is that there is an inherent standardization of the operations that can be applied to the resources, without needing to explicitly define descriptions of the methods that can be applied to them using the web services description language (WSDL). The URL contains all the information that is needed to return to a particular state of a web service. By including the name of the sensor, the temporal scope, and the filter parameters all in the URL, stream feeds make it easy to link to stream feeds, or even to link to partial stream feeds.

5 Implementation of the Stream Feed Abstraction

Our motivating application, MetroNet, was implemented using the Ruby on Rails framework. MetroNet is served using the lighttpd web server with the FastCGI module, and the MySQL database. The ActionController under Rails handles the complex routing requests required by StreamFeeds, so that XML feeds can be generated on demand based on the URL structure.

In MetroNet, all data is stored in a central database and a XML file is generated on the fly for each query. Thus, queries over historical data are performed as normal HTTP queries, and the generated file is returned to the client. Filters over historical queries are executed when the XML file is first generated.

Queries over incoming data are currently handled through a publish/subscribe mechanism. Every stream holds a list of subscriptions, and a client can add a receiver address by posting a URL to the location

http://.../sensorName/subscription

using the HTTP PUT command. Incoming data is then sent using a POST to the target URL. This interface is slightly different than that described in Section 4, although it achieves the same goals. The interface implementation is currently being changed to match that of Section 4.

The implementation described above is not fundamental to the steam feeds abstraction, and other implementations are possible as long as they provide the

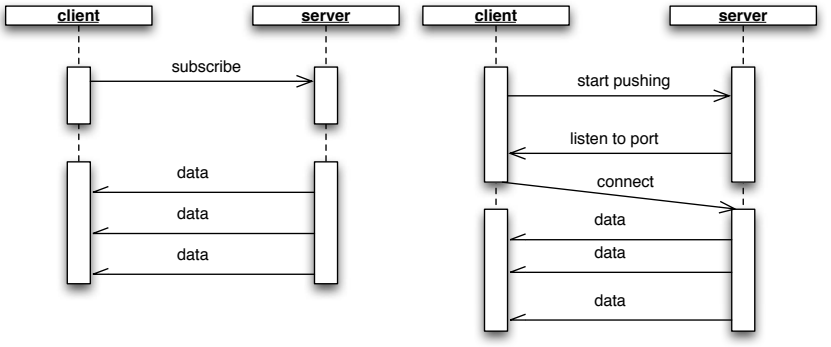


Fig. 4. A publish/subscribe mechanism can be used to efficiently push aperiodic to the client (left) while a new connection can be opened and remain open to push data to a client behind a firewall (right)

same interface and abstraction. For example, a stream feed could be stored in a flat file and served by a normal Web server. Filters could be implemented using syntactic filtering of entries with XML tags that match the filter. Incoming data could be appended to the file by a program that is independent of the Web server software, and the server could server queries over incoming data by observing changes to the file.

One limitation of the publish/subscribe implementation that we describe above is that clients cannot be behind firewalls or network address translation (NAT) devices. An alternative implementation is to negotiate a new long-lived TCP socket between the client and server, and to serve all HTTP responses over that socket. This scheme is compared with the protocol that we implemented for MetroNet in Figure 4. It would be able to stream data to clients behind NATs and firewalls, but has the disadvantage that a socket must be open during the entire query response, which could limit the number of queries that can be supported by the server at a time.

6 Providing Other Interfaces for Stream Feeds

The URL interface is not the only type of interface that can be provided to steam feeds. The URL interface is necessary to make sensor streams first-class citizens on the Web because they can be linked to and they can be crawled and indexed by search engines. However, we can also provide other interfaces by wrapping the URL-based interface. for example, we implemented the interface shown in Figure 1 using SOAP objects, providing a programmatic interface to stream feeds that can be used in any language. In our MetroNet server, the SOAP object for each stream can be located at a URL relative to the location of the stream itself. For example:

```
/streams/frontdoor/temperature/service.wsdl
```

The SOAP interface we provide contains three functions corresponding to the three temporal scopes over which queries can be posed. It also contains a member variable for each TAG in the XML schema used for that stream, so that the programmer can define a filter based on that XML schema. The filter defined through these variables is applied to all queries that are posed with the SOAP object, with any temporal scope. This interface provides the same abstraction as our URL-based interface and therefore provides the same abstraction over a sensor stream, but has the disadvantage that it is not as easily crawlable. Other similar APIs could be defined for stream feeds as necessary by wrapping the URL-based interface.

Table 1. A SOAP interface provides the same functionality as our URL-based interface in a programmatic way. Other interfaces can also be created for stream feeds.

API	Meaning
get()	query over historical data
getHistorical()	query over incoming data
getIncoming()	query over incoming data
TAG	a member variable through which to apply filters
TAG.eq	return only content where TAG equals a value
TAG.ne	return only content where TAG does not equal a value
TAG.lb	return only content where TAG is greater than a value
TAG.ub	return only content where TAG is less than a value

7 Evaluation

We evaluate the efficiency of the stream feed abstraction by measuring the scalability of the stream feed abstraction when applied to a query over a deep fusion tree, and compare our results from using standard Web feeds. We perform this evaluation in two parts. First, we empirically evaluate our server implementation to establish the trade-off between posing repeated queries for historical data and posing a single query for incoming data. Then, we use these empirical results to drive a simulation in which we pose queries over large fusion trees and measure the latency.

7.1 Repeated Historical Queries vs. Incoming Queries

In our first experiment, we empirically decide between repeatedly querying for the sensor stream and pushing the sensor stream. We setup a Web server that serves a sensor stream that is updated with a frequency of 1Hz. We also setup a set of clients that must receive the data with a frequency from 0.1Hz to 10Hz. The clients filter the query response by indicating the sequence number of the last value received, so only new data is sent in response by the server. We measure the total number of streams that can be served when all clients use push and when all clients use pull. The total values averaged over 100 runs can be seen in Figure 5.

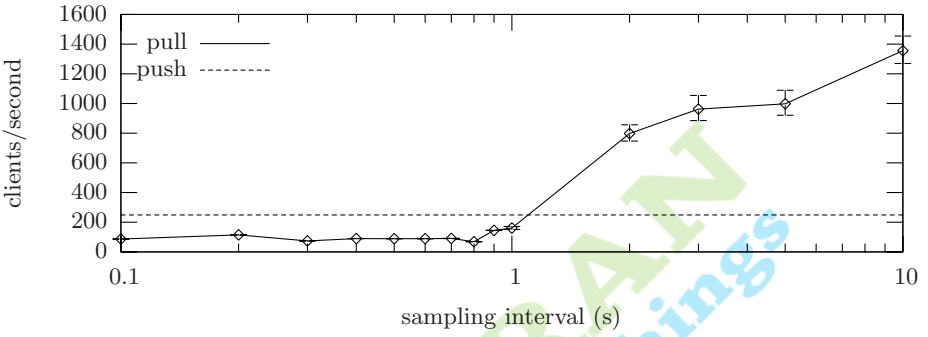


Fig. 5. When clients pull data from the server, the number of clients that can be served increases as the the frequency of requests increases. At a query rate of approximately 1 second, the number of clients served begins to exceed that supported when the server pushes data to the clients in real time.

In our experiment, we used the SOAP implementation to measure the performance of the query over incoming data, because the SOAP implementation keeps a socket open for the duration of the push, just as our proposed abstraction would due to problems with firewalls and NATs.

The results show that in the server we implemented, the PUSH queries can support a total of 250 clients at a time, independent of the frequency with which the client needs the data. This is because our web server can only support a maximum of 250 sockets at a time, and the PUSH query keeps the TCP socket open the entire time. This number may differ from servers to servers, due to the capability of the servers.

Theoretically, if each query over historical data only takes 0.2 seconds on the server side, then the server can support a total of more than $5 * 250$ queries in one second because the socket connection of a PULL request lasts less than 0.2 seconds. However, during the experiments we found that the TCP connection number is not the only factor that limits the number of clients we can support per second. As the query frequency increases, with 200 clients, the latency for each query increases due to the speed of database access. For queries over incoming data, a database access only occurs once when any new reading of the stream comes in. Then the server duplicates the new value and pushes it out to all clients. For repeated historical queries, on the other hand, each query requires access to the database because the server doesn't know whether the value of that sensor stream has changed since the previous client's query. This increases the latency of each query and brings down the number of queries we can support per second.

Our results show that more streams can be served using repeated historical queries if the query frequency is less than 1.2Hz. Therefore, since the data was being generated in this experiment at a frequency of 1Hz, we empirically validate that in our system the client should use running queries over incoming data when

the sampling frequency is 1.2 times the data generation frequency or more, and should use repeated historical queries otherwise.

7.2 Scalability and Efficiency

In our second experiment, we use the results from our first experiment to test the efficiency of stream feeds when used with deep fusion trees in simulation.

With each run, we randomly generate a network of servers with a branching factor of 3 as the depth increases. Although the network resembles a tree, some nodes can have multiple parents assigned to random nodes in the layer above. The nodes are connected in multiple layers from 2 to 15 as shown in Figure 6. The nodes on the bottom are sensors and generate data which is requested by several clients through a series of servers. 5 clients are attached to each server node, and feed the network with queries. The clients are initialized with a random query frequency, and each node has a query rate equal to the sum of the parent's query rates. The latency results reported are the averaged for every client in the network over 5 runs.

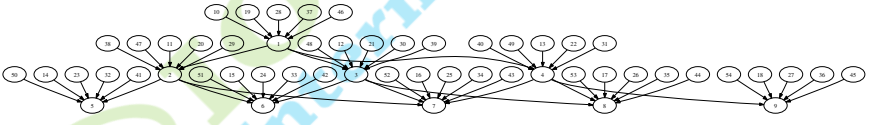


Fig. 6. A randomly generated *fusion tree* of 54 nodes randomly distributed and connected in three layers, with five clients per server

In our first case, we demonstrate the standard web feed abstraction by restricting servers only to pull for data without server-side filtering from a fusion tree of varying depth. The results from Figure 7 show that the average latency for each query grows exponentially with the depth of the tree. This is a problem, and demonstrates that normal Web feeds cannot efficiently support deep fusion trees.

In our second case, we enable servers to use filtering to prevent responses from returning results that are already known by the client. In other words, each client stores the sequence number of the last retrieved data point and passes that as a filter to the server in each request. This experiment shows that the latency grows proportionally with the depth of the network because the filtering significantly reduces the network traffic among all layers during the fusion process, and the latency is dominated by the roundtrip time rather than the transmission time.

In our third case, we evaluate the model with the two streamfeeds extensions, filtering and the ability to push. Each server dynamically decides between requesting historical queries or running queries over incoming data based on its observed query rate and data generation rate. We expect this optimization, by combining the advantages of pushing and pulling, to reduce the average latency of all queries even further than the reduction observed from filters. Because of this dynamic switching there is a balance point at a certain depth in the network, where all the nodes beneath that level begin to push their data to the

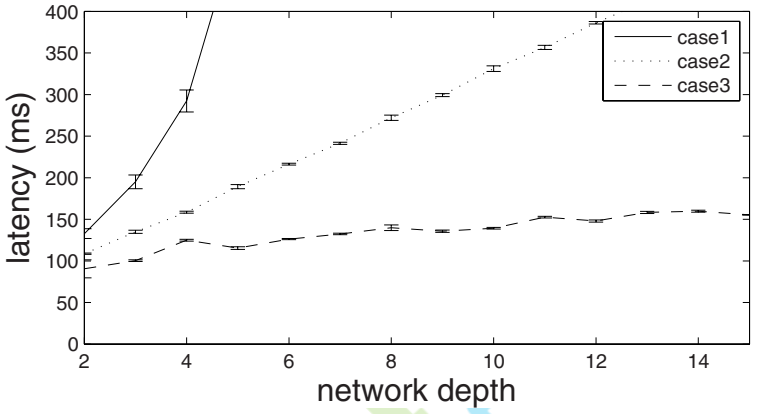


Fig. 7. As the network gets deeper, filtering with the ability to push outperforms the alternatives

parents. Because of this region of low latency, the queries suffer most latency when pulling for data in the upper levels of the topology, however more clients can be served in these upper levels. The result is an improvement in the the average latency for all the client's requests. We expect that with larger files, lower sampling rates, and/or deeper fusion trees, this benefit would be further increased.

8 Conclusions

In this paper, we showed that we can provide an efficient abstraction for sensor streams that is a first class citizen on the Web by combining the advantages of Web feed and multimedia streaming paradigms. Each sensor stream has a URL, and parameters encapsulated in this URL allow the user to query for historical data using a PULL model, to query for incoming data using a PUSH model, or both. Other parameters to this URL allow the user to filter the data on the server side, to reduce the size of the data response in repeated queries to the large streaming data objects.

Because this abstraction is a first-class citizen on the web it inherits security, authentication, and privacy mechanisms. Stream feeds can also be crawled and indexed by search engines. The servers are all autonomous, so the server administrators do not need to worry about storing data on the servers of others, or having other data stored on this domain's servers, as with the IrisNet and Hourglass systems. Thus, in this paper, we show that the stream feed abstraction meets the requirements of and can provides the basis for the World Wide Sensor Web, where sensor data can be shared, processed, and published within a single global framework.

References

1. Page, L., Brin, S.: The anatomy of a large-scale hypertextual web search engine. WWW7 Computer Networks 30, 107–117 (1998)
2. Gormon, B., Shankar, M., Smith, C.: Advancing sensor web interoperability. Sensors (April 2005)
3. Geller, T.: Imaging the world: The state of online mapping. IEEE Computer Graphics and Applications 27(2), 8–13 (2007)
4. Abrrer, K., Hauswirth, M., Salehi, A.: A middleware for fast and flexible sensor network deployment. In: VLDB 2006: Proceedings of the 32nd international conference on Very large data bases, VLDB Endowment, pp. 1199–1202 (2006)
5. Rose, I., Murty, R., Pietzuch, P., Ledlie, J., Roussopolous, M., Welsh, M.: Cobra: Content based filtering and aggregation of blogs and rss feeds. In: 4th USENIX Symposium on Networked Systems Design and Implementation (2007)
6. Schulzrinne, H., Rao, A., Lanphier, R.: Real time streaming protocol (rtsp). Technical report, Network Working Group (1998)
7. Saint-Andre, P.,, E.: Extensible messaging and presence protocol. Technical report, Jabber Software Foundation (October 2004)
8. Hickson, I.: Html 5. Technical report, Google (September 2007)
9. Armenio, F., Barthel, H., Burnstein, L., Dietrich, P., Duker, J., Garrett, J., Hogan, B., Ryaboy, O., Sarma, S., Schmidt, J., Suen, K., Traub, K., Williams, J.: The epcglobal architecture framework. Technical report, EPCglobal (2007)
10. Cooper, O., Edakkunni, A., Franklin, M.J., Hong, W., Jeefrey, S., Krishnamurthy, S., Reiss, F., Rizvi, S., Wu, E.: Hifi: A unified architecture for high fan-in systems. In: Proceedings of the 30th VLDB Conference (2004)
11. Gibbons, P.B., Karp, B., Ke, Y., Nath, S., Seshan, S.: Irisnet: An architecture for a world-wide sensor web. IEEE Pervasive Computing 2(4), 22–33 (2003)
12. Huebsch, R., Hellerstein, J.M., Boon, N.L., Loo, T., Shenker, S., Stoica, I.: Querying the internet with pier (2003)
13. Cherniack, M., Balakrishnan, H., Balazinska, M., Carney, D., Cetintemel, U., Xing, Y., Zdonik, S.: Scalable Distributed Stream Processing. In: CIDR 2003 - First Biennial Conference on Innovative Data Systems Research, Asilomar, CA (January 2003)
14. Chandrasekaran, S., Cooper, O., Deshpande, A., Franklin, M.J., Hellerstein, J.M., Hong, W., Krishnamurthy, S., Madden, S., Raman, V., Reiss, F., Shah, M.: Telegraphcq: Continuous dataflow processing for an uncertain world (2003)
15. Chen, J., DeWitt, D.J., Tian, F., Wang, Y.: NiagaraCQ: A scalable continuous query system for Internet databases, ACM SIGMOD, pp. 379–390 (2000)
16. Liu, S.N.J., Zhao, F.: Challenges in building a portal for sensors world-wide. In: First Workshop on World-Sensor-Web: Mobile Device Centric Sensory Networks and Applications (WSW 2006) (2006)
17. Shneiderman, J., Pietzuch, P., Ledlie, J., Roussopolous, M., Seltzer, M., Welsh, M.: Hourglass: An infrastructure for connecting sensor networks and applications. Technical report, Harvard University (2004)

Author Index

Adamer, Kurt 289
Åkesson, Karl-Petter 263
Alonso, Gustavo 87
Anson, Omer 68

Bannach, David 289
Baus, Jörg 247
Bay, Herbert 230
Beigl, Michael 155
Berchtold, Martin 155
Beuster, Monty 155

Choi, Jongmyung 276
Condea, Cosmin 19
Coronel, Pedro 325

Dada, Ali 140
de Souza, Luciana Moreira Sá 50
Decker, Christian 155
Dickerson, Robert 360
Dörr, Robert 247
Duller, Michael 87

Evdokimov, Sergei 1

Fabian, Benjamin 1
Fahlén, Lennart E. 263
Furrer, Simeon 325

Gilmer, Ken 87
Glezer, Chanan 68
Grummt, Eberhard 35
Guinard, Dominique 50
Günther, Oliver 1
Guo, Xin-Can 184

Hansson, Pär 263
Havinga, Paul 341
Herzig, Daniel 155
Herzog, Thomas 155

Jedermann, Reiner 105

Karlgren, Jussi 263
Karnouskos, Stamatis 50
Kasten, Oliver 19
Klug, Tobias 289
Köhler, Moritz 50
Krishnamurthy, Sudha 68
Kürschner, Chris 19

Lang, Walter 105
Liu, Hsin-Chin 184
Lombriser, Clemens 341
Lopez-de-Ipina, Diego 308
Lu, Jiakang 360
Lu, Jian 360
Lukowicz, Paul 289

Maragkos, Damianos 87
Marin-Perianu, Raluca 341
Mitsugi, Jin 173
Müller, Markus 35
Murai, Jun 173

Nakamura, Osamu 173

Osaka, Kosuke 173

Papageorgiou, Dimitrios 87

Quack, Till 230

Rellermeyer, Jan S. 87
Riedel, Till 155
Roehr, Daniel 155
Rois, Mauro 68
Ronzani, Daniel 214
Rothensee, Matthias 123

Sapir, Lior 68
Savio, Domnic 50
Sbodio, Marco Luca 289
Schloeder, Kilian 68
Schmid, Viola 196
Schmitz, Michael 247
Scholten, Hans 341
Schott, Wolfgang 325
Shub, Ilana 68

IOT IPV Internet of things

Söderberg, Jonas 263
Spiess, Patrik 50
Ståhl, Olov 263

Thiesse, Frédéric 19, 140
Tröster, Gerhard 341
Tresman, Mimi 289

Van Gool, Luc 230
Vazquez, Juan Ignacio 308

Wallberg, Anders 263
Weiss, Beat 325
Weiss F. Chaves, Leonardo 155
Whitehouse, Kamin 360

Yoo, Chae-Woo 276

Ziegert, Thomas 289
Zinnen, Andreas 289

IOT IRAN
Internet of Things