CREATING AD-HOC PERVASIVE COMPUTING ENVIRONMENTS

Michael Beigl, Tobias Zimmer, Albert Krohn, Christian Decker, Philip Robinson

1. Introduction

Pervasive Computing technology can convert everyday environments, such as office spaces, into digitally enriched places. Users can profit from pervasive computing environments in many ways: context aware applications may proactively react [4], leading to, for example, information being displayed based on the user's current location. In bringing more and more of these technologies together, the potential will rise yet the effort to keep these systems running will increase. Today many approaches require a pre-installation of technology and maintenance of the setting. In our vision such environments will not be planned and installed by experts, but will emerge or even be assembled ad-hoc. This will enable users, unfamiliar with computing technology in general and pervasive computing technology in particular, to profit from such environments without the need for administration and maintenance by others.

This paper introduces Impromptu, a concept that enables us to build pervasive computing environments from scratch without the need for infrastructure and without expert knowledge and configuration. Impromptu defines concepts and provides a framework with enabling technology for creating systems that can later be used in Impromptu environments. The Impromptu specifies a computing device that can be part of such an environment, while the Impromptu framework provides technology that supports building technology for Impromptu environments.

2. Impromptu Concept

Impromptu environments are built up by naturally introducing everyday objects that are enhanced with computing, sensing and communication technology. Everyday objects can be either typical consumer appliances, such as computing devices, PDAs, Digital Cameras, or they can be everyday objects that are post-hoc converted into appliances using e.g. Smart-Its [5] technology. The aim of Impromptu is to create environments where computing-based applications are able to profit from situation information that is detected by some or all of the participating devices and subsequently communicated between those devices. In the concept this should be done without interfering with the user or forcing him to set-up, configure, maintain or administer such environments.

From experiments within our experimental office environment we found that devices should fulfill three conditions to make Impromptu environments possible:

- Be able to detect external and/or internal context information

- Be able to communicate this information to all other devices in the environment by using the same high-level communication language

- Be able to work without configuration and administration

3. Impromptu Framework

Computing systems that follow the above rules are suitable to take part in an Impromptu environment. To make development of such devices simpler or to enable users of off-the-shelf computers to take part in an Impromptu environment, we developed the Impromptu framework. Core elements of this framework are:

- Smart-Its *Particles* hardware and software technology [2] to provide computing, networking and perception capabilities to everyday objects and environments.

- The *ConCom* (Context Communication) Language, which is part of the Aware-Con communication Network [1]. This communication language allows computing devices to communicate to any Impromptu enabled computing system and to understand communication from these systems

- A standard way for *Context Perception* which is needed to perceive the world through sensors and communication

- Off-the-shelf computers – *laptops or PDAs* - and *appliances* that are integrated in our environment using add-on services

3.1. Particle Computer

Smart-Its Particles are small computing devices that are able to perceive the surrounding environment and their internal state, to process information and to communicate this and other information to peers in the environment. They are intended to be used in settings where small size and low power consumptions matter. Particles can be used to post-hoc add computing technology to existing everyday objects as chairs, tables or pens. Particles are embedded in or adhered to such an object and hence retrieve context information via the integrated sensors (figure 1). Another option to use Particles is to add them to new or existing electronic devices by utilizing the various interface possibilities.

Particles consist of basic hardware technology such as sensors, actuators, communication, processing and memory and come with software supporting context recognition and communication. Particles can either be used with pre-installed programs or by using the software library and operation system. Both hardware and software is intended to form a maintenance and administration free appliance. Particles include the ConCom high level communication language interface that enables the device to understand communicated information from other devices and to send information to peers. Derivation of context information is made possible by the use of our sensors and context API.

Figure 1 Particle attached to a table

3.2. ConCom and Network Integration

The Context Communication (ConCom) language allows participating systems to communicate to each other directly in an ad-hoc manner. By defining the types and structure of communication ConCom builds the basis for a common understanding between participating systems. ConCom is especially build to support the exchange of context communication and communication of data with added context.

In ConCom all communication is packed into a list of tuples that form a sentence. A sentence always starts with a type identification of the object sending the information as "Artefact: I am a pen" or "Artefact: I am a doorplate". The sentence is continued by a list of additional information like "Context: Someone is writing with me" or "Context: Situation Meeting". Tuples are typed and basic types - Artefact and Context – that are known to each of the parties taking part in the communication. Devices may interpret all or only part of the information of one sentence. This way it is possible to introduce new types of devices providing additional information without tampering the functionality of existing devices.

While Particle computers can speak to each other instantly using the built-in API, other computing devices like laptop computers, PDAs or mobile phones, can also be enabled to take part in the communication by using software add-ons. They can therefore talk and understand ConCom through one of their network interfaces including Ethernet, WiFi or Bluetooth. Bridges build the basis for physically and logically interconnecting various types of networks including the AwareCon network used by the Particles. Logical information flows are routed using the RAUM location information [3], which is embedded into each packet.

3.3. Context System and Perception

The added value of the system comes from the new abilities of the objects with their embedded computing devices: The ability to perceive their own situation and the situation around. For example, a whiteboard pen can now recognize that it is used for drawing something – or even what it draws, if someone is playing with the pen or if the pen is not used at all. The context information within the pen is derived in several steps from raw

sensor data by adding domain knowledge such as the type of sensor that is implemented and the type of object. So the pen derives the state of usage like: “writing”, “playing” and “laying down” from its raw motion pattern. The high level information is then transferred to other devices in the environment. Devices in the environment such as a digital camera pick up the information of the pen. The devices join this information with other information, as the “this-is-a-meeting” context provided by other devices, and derive a higher-level context. This information is then used to trigger application functions.

4. Conclusion

Impromptu has enabled us to rapidly build pervasive computing environments. Our first test environments were office spaces, where we equipped everyday objects with Particles computing technology and integrated them into settings with laptop, PDA devices etc.. ConCom as the communication language was found very useful in these settings as it allowed distributed functionality across various platforms. It was also very helpful in allowing standard off-the-shelf computing devices to receive situation information from surrounding sensor devices. The Impromptu concept helped us to make applications completely configuration free and independent of administration.

5. References

[1] Michael Beigl, Albert Krohn, Tobias Zimmer, Christian Decker, Philip Robinson
AwareCon: Situation Aware Context Communication, Ubicomp 2003, Oct. 12-15, Seattle, USA

[2] Beigl, M., Zimmer, T., Krohn, A., Decker, C., Robinson, P.: Smart-Its - Communication and Sensing Technology for UbiComp Environments. Technical Report ISSN 1432-7864 2003/2

[3] Beigl, M., Zimmer, T., Decker, C.: A Location Model for Communicating and Processing of Context. Personal and Ubiquitous Computing Vol. 6 Issue 5-6, pp. 341-357, ISSN 1617-4909, 2002

[4] David Tennenhouse, Proactive Computing, Communications of the ACM, Volume 43, Issue 5, May 2000

[5] The Smart-Its Project Web Site: http://www.smart-its.org (visited 02/2004)