Publication IV

I. Jantunen, H. Laine, V. Ermolov, and P. Huuskonen. Smart sensor architecture for mobile terminal centric ambient intelligence. In *Eurosensors 2006*, (Göteborg, Sweden), vol. 2, pp. 294–295, Sep. 2006.

SMART SENSOR ARCHITECTURE FOR MOBILE TERMINAL CENTRIC AMBIENT INTELLIGENCE

Iiro Jantunen[1], Hannu Laine[1], Vladimir Ermolov[1], Pertti Huuskonen[2]

[1] Nokia Research Center, Helsinki, Finland, [2] Nokia Research Center, Tampere, Finland

*Corresponding author: Iiro Jantunen, +358504868699 (TEL), +358718063214 (FAX), iiro.jantunen@nokia.com

Abstract: This study is about making ambient intelligence reality by developing a mobile phone centric open architecture for sensor networks. The architecture is developed and demonstrated for using new sensors, and new mobile services and applications. The technologies used in the architecture include Simple Sensor Interface (SSI) protocol, nanoIP and low-power short-range radios.

Keywords: smart sensor architecture, wireless sensors, remote sensing

INTRODUCTION

Various architectures have been developed for wireless sensor networks. In this paper we describe a sensor network architecture at which a mobile phone acts as the trusted intelligent user interface for smart sensor networks (see Figure 1).

Compared to other candidates for a user interface device (laptop computers or PDAs), mobile phones have several advantages: highest penetration and acceptance amongst users, relatively low cost and small size, both local and long range wireless connections from everywhere to everywhere, access to a wide range of services via internet, data storage possibility and local computational capacity, and that no additional user interfaces need to be carried by the user.

This architecture has been developed in frame of EU 6th framework programme project MIMOSA. The architecture is referred to as MIMOSA architecture in this paper.

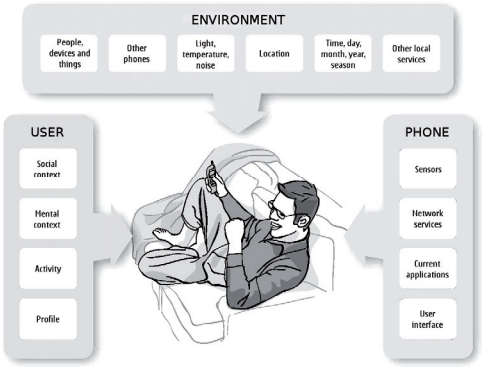

Figure 1. Mobile phone as the central device of sensor network

MIMOSA ARCHITECTURE

The MIMOSA sensor architecture is designed to be modular and freely scalable. The mobile phone (terminal device) has open software interfaces for 3rd parties through Context application programming interface (API), Sensor API, and Local Connectivity API. Open interfaces for 3rd party sensor hardware are also provided.

MIMOSA architecture defines four types of entities: **terminal devices** with built-in sensors, **sensor radio nodes**, **RFID sensor tags** (radio frequency identification) , and **back-end servers**. The terminal device provides, in addition to cellular network connection, Bluetooth, BluLite and RFID radio interfaces. Back-end servers are computers providing data storage, data processing and extra services.

Sensor radio nodes are wireless battery-powered smart sensors running sensor server software. RFID sensor tags are passive RFID tags with a sensor.

BluLite is a radio technology specially designed for sensor networks [1]. Sensors could add value by having wireless radio connection to mobile terminal but cannot bear the power consumption and cost associated to Bluetooth. BluLite tackles the mismatch by introducing minor power-saving additions to the Bluetooth chip. It is relatively cheap to provide BluLite chips or add the functionality to Bluetooth chips – in contrast to adding other radio technologies designed for sensors, e.g., ZigBee.

Plug-in type implementation of sensors is the key to modularity. The Sensor API will detect what sensors are available regardless their location in terminal, in sensor radio node, or in RFID tag. The Sensor API on the host device will keep a list of available sensors and provide functions for accessing the sensors, be they local (connected directly to the host device) or remote (RFID or BluLite connected).

MIMOSA software architecture is based on software layers whose APIs are accessible to 3rd party software.

The **Context layer** abstracts sensor-level data into higher level units, so called "context atoms". For instance, temperature and humidity measurements can be converted into dew point measurements, or heart beat data can be converted into subjective "low", "normal", "high" classes. The layer achieves this through a rule-based reasoning mechanism and an API based on publish-subscribe model.

Sensor layer provides a SSI server, a RFID Sensor Tag reading interface and a Sensor API for

3rd party software. Sensor layer finds and reads both locally connected and wireless sensors.

Simple Sensor Interface (SSI) protocol [2] defines a method for reading sensors regardless their type, location or connection between the sensor and the reader. SSI is client-server architecture, with sensor devices acting as SSI servers and terminal devices act as SSI clients. A single sensor device can have multiple sensors. Both polling sensors by client terminals and streaming data from sensor servers are supported.

SSI protocol is an application level protocol that can be used over any network environment.

Local Connectivity Layer provides a Local Connectivity API to read locally connected, BluLite and RFID sensors. Connection to MIMOSA hardware is over a SPI bus with extra interrupt lines (MIMOSA SPI, M-SPI). BluLite sensors are networked with nanoIP [3, 4].

ARCHITECTURE IMPLEMENTATION

The proposed architecture (see Figure 2) is demonstrated with four different hardware entities: terminal device, sensor radio node, RFID sensor tag, and back-end server.

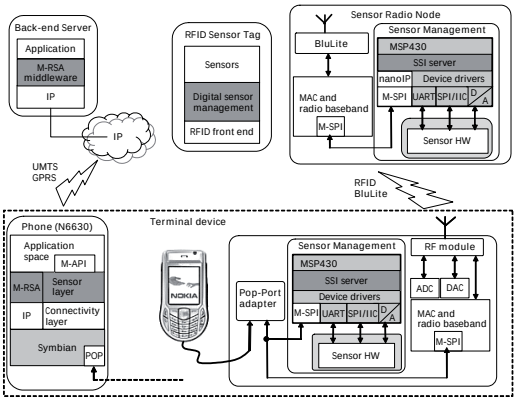

Figure 2. Architecture implementation

Sensor management is done with a MSP430 microcontroller which runs an SSI server, nanoIP networking, and sensor drivers. There is also a real time clock and a standard connector for sensor hardware. Sensor management provides a power supply for the sensor hardware. The same sensor management unit is used in both Sensor Radio Node and Terminal Device, the difference being in communications software.

The sensor unit can either be an analog sensor with amplified signal or a digital sensor.

Terminal device is a 3G mobile phone, a Nokia 6630 (Series 60), with add-on electronics and software layers to provide MIMOSA functionality. The mobile phone acts as a gateway between local sensor environment and internet, where remote servers can provide extra functionality.

There are several software layers for additional software to connect to (see MIMOSA architecture). The phone provides Bluetooth, BluLite and RFID radios for reading sensors.

Sensor Radio Node is a wireless battery-powered smart sensor with BluLite radio, nanoIP networking, and SSI server software. SSI server is implemented on a sensor management MSP430 microcontroller.

RFID Sensor Tag is a wireless remote-powered sensor designed for low-cost sensing. The reading signal powers the sensor which then writes the sensed value to its memory, which in turn is read by the Sensor API of the Terminal Device.

RFID Sensor Tag is a Mode 1 (passive backscatter RFID system) tag as defined by ISO 18000-4 standard. SSI protocol specification defines the memory layout of RFID Sensor Tags. In MIMOSA architecture, RFID Sensor Tags work on the same frequency as BluLite, 2.4 GHz, to remove the need of several radio front-ends.

As a **Back-end Server**, a laptop computer with Mimosa Remote Sensing Architecture (M-RSA) middleware in another location was used to read sensor data over IP.

RESULTS AND DISCUSSION

MIMOSA architecture has been successfully demonstrated in laboratory environment. Reading built-in sensors and sensors on Sensor Radio Node has been demonstrated. The read data is used by applications both in mobile phone and in back-end server.

CONCLUSIONS

After further development the proposed architecture can be used as an open architecture platform for implementing for mobile phone centric ambient intelligence in different application areas.

ACKNOWLEDGEMENTS

This study is part of MIMOSA project under EU FP6 contract IST-2002-507045.

CEA-LETI has provided the RFID Sensor Tag. VTT has provided the RF block for the RF module.

REFERENCES

1. M. Honkanen, A. Lappetelainen, K. Kivekäs; *Radio and Wireless Conference 2004,* IEEE, 19-22 Sept. 2004, pp. 199-202.
2. http://ssi-protocol.net
3. Z. Shelby, P. Mahonen, J. Riihijärvi, O. Raivio, P. Huuskonen; *IEEE International Conference on Communications, 2003. ICC '03*, 11-15 May 2003, Vol. 2, pp. 1218-1222.
4. http://www.cwc.oulu.fi/nanoip/