A PROTOTYPE AMBIENT INTELLIGENCE SENSOR NETWORK BASED ON A COMPUTER AND RADIO TRANSCEIVER.

D.A. Sanders, CR Smart, A Gegov and B Gremont

University of Portsmouth, Portsmouth, PO1 3DJ, UK

E-mail: *david.sanders@port.ac.uk*

Abstract

This paper describes a prototype open architecture for implementing an ambient intelligence radio system. In the proposed sensor network, a radio transceiver connects a local sensor network to a server across the internet. A context awareness layer abstracts sensor measurements through rule-based reasoning and notifies changes to local and remote applications. Low-power short-range radios are used.

Introduction.

Wireless sensor networks are usually are ad-hoc peer-to-peer networks requiring computational and networking capacity on sensor nodes (Chong, 2003). A radio transceiver acts as the user interface to simplify development. A computer connected to the transceiver provided local network and internet connections.

The computer used the radio transceiver to search and read sensors and forwarded data to servers across the internet. The wireless sensors were small and battery powered.

After some background about ambient intelligence, the key components and software layers are introduced next, followed by some notes on implementation and functionality.

The work is based on work by Iiro Jantunen with Nokia mobile telephones at the Nokia Research Center in Helsinki (Jantunen *et al*, 2007). Instead of telephones, simple radio connections were used.

Background

Early developments in Ambient Intelligence took place at Philips (Sanders and Gegov, 2006). In 1998, Philips commissioned a series of internal workshop to investigate different scenarios that would transform the high-volume consumer electronic industry from the current "fragmented-with-features" world into a world in 2020 where user-friendly devices support ubiquitous information, communication and entertainment. In 1999, Philips joined the Oxygen alliance, an international consortium of industrial partners within the MIT Oxygen project (Oxygen, 2006) aimed at developing technology for the computer of the 21st century. In 2000, plans were made to construct a feasibility and usability facility dedicated to Ambient Intelligence. This opened in 2002. Along with the build up of the vision for Philips, a parallel track was started to open up the vision. Following the advice of the Information Society and Technology Advisory Group, the European Commission used the vision for the launch of their sixth framework (FP5) in Information, Society and Technology and the Commission played a crucial role in further developing Ambient Intelligence. More recently, several major initiatives have been started in the USA, Canada, Spain, France and the Netherlands.

Architecture

The computer with the transceiver could process data from sensors and could connect to servers and other computers to store data. The architecture was designed to be modular (Rogers *et al*, 2006) and was based on three levels (Tewkesbury and Sanders, 1999): a. Context, b. Sensor, and c. Local Levels.

a. Context Level. The Context Level took the data form the sensor-level and converted it to useful information. The most recent sensor value was stored with relevant environmental data. The Context Level monitored changes and notified applications of the changes. Data at the Context Level was held in a central data structure modelled on the blackboard paradigm described by Hudson (Hudson *et al*, 1997). Data on the blackboard was visible to all interested agents and local systems.

Computer applications could upload and remove data and the Context Level monitored changes to sensor values and environmental values.

b. Sensing Level. The sensor level provided sensor data selection, acquisition and delivery to the servers. The level provided remote communication between the sensors. The sensor level provided an interface to all the sensors regardless of type, location or connection. An interface checked access rights and verified the time that sensor data was last acquired and whether a sensor was available to provide real time data.

c. Local level. Sensor data was obtained for future use and stored with a local storage component. The Local Level sent and received messages to and from sensors (and other devices) by radio. The local level provided a sensor client, an RFID sensor tag reading interface and a Sensor API for upper level or 3rd party software. The layer found and read both locally connected and wireless sensors.

The server

The server dealt with requests for sensor data and results were delivered back to the Context Level.

Apart from data acquisition, discovery constituted another big role of the middleware, enabling applications to query the availability of certain sensors and aggregate functionality. A registration and availability component implemented the necessary functions.

The application server's registration and availability component serves as a central registry for all registered gateways. After storing the obtained sensor information locally (for use with gateway-local applications), the information was published at the application server's registration and availability component.

The available information in the registration and availability component in the application server could either be queried by the application directly or by the acquisition component before sending an acquisition request to the intermediary gateway. Apart from a simple discovery request, the component also provided functionality to subscribe to the availability of sensors or aggregation functionality. Hence, applications were notified when such information became available.

To speed up data transfer in streaming, the server buffered data points. The number of data points that could be sent in a single message depended on the buffer length, which was 128 bytes in this work, resulting in a limit of 29 data points, each 4 bytes.

Prototype architecture

The prototype architecture was based on (Jantunen *et al*, 2007). The Terminal Device was a computer with a simple radio transceiver. The computer ran local sensor applications, and acted as a gateway between the local sensors and internet, where servers could provide extra functionality.

The connection from computer to radio transceiver was over an USB to SPI converter (Pop-Port adapter). Add-on sensors were interfaced and managed by a sensor management board of the same kind as in the Sensor Radio Node. The only difference was in networking, which was done by point-to-point protocol.

The sensor was a wireless battery-powered smart sensor with server software. Sensor management was achieved with a microcontroller running the server, networking, and drivers for sensor and communications hardware. There was also a real time clock, which could also be used as a time sensor.

The sensor management board provided a standard connector for sensor hardware, providing a power supply, and including standard digital interfaces, analogue input lines and configurable general digital input/output lines. The latter could be used, for example, to implement a non-standard digital interface to the sensor(s), or provide sensor enable or clock signals.

The server was actually a Sony Vaio laptop computer in another location. That was used to read sensor data over the network. The terminal device acted as a link that forwarded sensor readings from the sensors to the computer.

Results and discussion

The prototype ambient intelligence sensor network based on a computer and radio transceiver was successfully demonstrated in a laboratory environment with a wireless sensor measuring temperature and pressure. The remote sensing application on the terminal forwarded selected values over the radio network and internet to a Sony Vaio laptop computer acting as a server.

The wireless sensor consisted of a connectivity board, a sensor management board with a microcontroller running device drivers and server software, and a sensor board. The sensors used custom digital interfaces to conserve energy.

The demonstration included searching for sensor devices, discovery of sensors and reading sensors across a radio link.

Conclusions and future work

The prototype ambient intelligence sensor network based on a computer and radio transceiver was optimized for flexibility and low-power use. The key entities included in the prototype architecture were terminal devices with add-on electronics to provide extra functionality, active radio sensors with their own power sources, passive sensors powered by the reading signal, and a server (lap-top). The computer provided the ability to run sensor applications and an internet connection and acted as a link between the sensors in and the server. The sensors could either be directly attached to the terminal device or linked via radio.

For the moment, the ambient intelligence in the system was mainly about the interactions between the human user and the system. Internal functioning of the software was not fully considered although the software was connected to the environment via the sensors.

The systems do consider user experience and more work is required to make the ambient intelligence more user-centred so that the user is placed nearer the centre of the activity rather than just receiving feedback from the sensors.

The distributed network could be made dynamic. The communication with the network is seamless but interoperability could be improved and they could be made to auto-configure to remove the need for software configuration programs to interface and begin operation.

References

Chong CY and Kumar SP (2003), Sensor networks: evolution, opportunities, and challenges, Proc. IEEE 91, pp. 1247–1256.

Hudson AD, Sanders DA, Golding, H *et al* (1997). Aspects of an expert design system for the wastewater treatment industry. Jnl of systems architecture, 43 (1-5), pp 59-65.

Jantunen I, Laine H, Huuskonen P, Trossen D and Ermolov V (2007). Smart sensor architecture for mobile-terminal-centric ambient intelligence. Sensors and Actuators A 142, pp 352–360

Oxygen (2006). http://www.oxygen.lcs.mit.edu

Rogers I, Tewkesbury GE, Powell W and Onuh S (2006). Creating a Modular Electronic Design system. Journal of computing in systems and engineering, 7 (2). Pp 83 – 92.

Sanders DA and Gegov A (2006). Ambient Intelligence. Journal of computing in systems and engineering, 7(1). Pp 78 – 82.

Tewkesbury GE and Sanders DA (1999). A new robot command library which includes simulation. Industrial Robot, 26 (1), pp 39-48.