Middleware for embedded sensors and actuators in mobile pervasive augmented reality

Pedro Miguel da Fonseca Marques Ferreira
Laboratory of Communications and Telematics (LCT)
Centre of Informatics and Systems of Coimbra University
Coimbra, Portugal
pmferr@dei.uc.pt

João Orvalho and Fernando Boavida
Laboratory of Communications and Telematics (LCT)
Centre of Informatics and Systems of Coimbra University
Coimbra, Portugal
orvalho@dei.uc.pt, boavida@dei.uc.pt

Abstract— **In this paper we describe a Java middleware, the SENSACT API, which enables the construction – and connection through a Bluetooth network – of a network of sensors and actuators for the purpose of enabling its use in mobile pervasive augmented reality games. The system enables sensors and actuators to be deployed and subsequently connected to a central coordinating device, the game device, that has a compatible middleware API, the Status Transmission Framework Version 2.0 PAN API. The presented middleware package was subject to functional as well as performance tests using Sun Microsystems Java Wireless Toolkit 2.3 Beta version, which showed that the proposed API is adequate for small sensor/actuator networks.**

Keywords- Pervasive networking, entertainment, augmented reality, ad-hoc networking, wireless networking

I. INTRODUCTION

A significant requirement of pervasive applications is fast service development and deployment [1], which implies the introduction of various service and application frameworks and platforms. For this, middleware is a common solution. The benefits of middleware utilization are the improved programming model, and the hiding of many implementation details, which make middleware based application development much faster.

It is now becoming quite clear that entertainment, and more specifically mobile gaming, will be one of the killer applications of future wireless networks [2].

Augmented reality extends reality with virtual elements while keeping the computer in an assistive, unobtrusive role [3]. It is possible to create games that place the user in the physical world through geographically aware applications. Most of the latest mobile phones are equipped with cameras and some of the latest ones are coming with some form of 3D rendering technology [4] [5]. Bluetooth technology and increasing miniaturization will lead, in the near future, to low-cost, specialized pervasive equipment for augmented reality.

In [6] we described the main objectives of our research concerning systems that satisfy the requirements of network middleware for large scale mobile and pervasive augmented reality games. In [7] we described a middleware system that is being developed for large scale mobile and pervasive augmented reality games that satisfies these objectives. The system targeted by the middleware is composed of 3 levels: the back-office central level, the large scale network level, and the personal area network level. This paper addresses the latter level.

II. PERSONNAL AREA NETWORK REQUIREMENTS

The personal area network (PAN) level consists of the network of pervasive devices dedicated to personal communications and to augmenting reality. These may be sensors, actuators, and other devices that interact using Bluetooth or other means of communication. All these communicate with the mobile host, most likely a cell phone or specialized device connected to a wide area network – the 3GPP network – thus enabling users to play augmented reality games irrespective of their location.

PAN devices must support the Java language, more specifically, Java Micro Edition, in its Connected Limited Device Configuration (CLDC) version 1.1, and the MIDP – Mobile Information Device Profile - version 2.0. They must also support the Java Bluetooth API (JSR-82), the Java SIP (Session Initiation Protocol) API for J2ME (JSR-180), and the location API for J2ME (JSR-189).

Other devices that are needed in the personal area network level are input and output devices. These must support CLDC 1.1 and the Bluetooth API. Output devices are essentially video and audio devices. Video and audio output devices should also support, besides CLDC 1.1 and Bluetooth for J2ME (JSR-82), the Mobile 3D graphics API (JSR-184), and the Mobile Media API for J2ME (JSR-135).

In order for any digital system to have an awareness of and be able to react to events in its environment, it must be able to sense the environment. This can be accomplished by incorporating sensors, or arrays of various sensors (sensor fusion) into the system. Sensors are devices that are able to take an analogue stimulus from the environment and convert it into electrical signals that can be interpreted by a digital device with a microprocessor.

For a sensor or array of sensors to be supported by our central coordinating device middleware – the status transmission framework [8] version 2.0 – and by the sensor or actuator middleware – the SENSACT API – it must be accompanied by hardware that has a compatible CLDC 1.1 and JSR-82 (Bluetooth for J2ME) implementations.

III. THE SENSACT API

We now describe the part of the system, the SENSACT API, which enables the sensors and actuators to be deployed with the help of java CLDC 1.1 and communicate by Bluetooth with the help of JSR-82.

The SENSACT API is a small footprint set of classes occupying less than 60KBytes, constituted by 5 java packages, with the functions shown in TABLE 1.

In order to deploy one sensor/actuator, one has to have the respective hardware, a CLDC 1.1 implementation that runs on the hardware, a Bluetooth API (JSR-82) implementation for that hardware and platform, and the SENSACT API.

TABLE 1 SENSACT API PACKAGES

pt.uc.dei.lcst.stf.sensact.main
Main package of the SENSACT API. Contains classes that can and should be used to create new sensors or new actuators.
pt.uc.dei.lcst.stf.sensact.pack
Contains a pack of discovery messages, sensor and actuator classes already implemented that greatly simplify building the most common types of sensors and actuators.
pt.uc.dei.lcst.stf.sensact.net
Creates a general channel architecture that can be used to abstract from the communication system, be it Bluetooth or other (in the future, for example, ZigBee could be supported by simply extending classes).
pt.uc.dei.lcst.stf.sensact.link
Implements communication channels and helper classes (Bluetooth)
pt.uc.dei.lcst.stf.sensact.persist
Extends persistence to the world of J2ME CLDC with a low bandwidth alternative to the standard Java 2 SE persistence classes.

IV. TEST RESULTS

The presented middleware was subject to extensive functional and performance tests, with various kinds of simulated sensors and actuators and a simulated reading and actuating application using Java Wireless Toolkit from Sun Microsystems running in a series of emulators in a Pentium 4 3.6 GHz System with 1 Gb Memory.

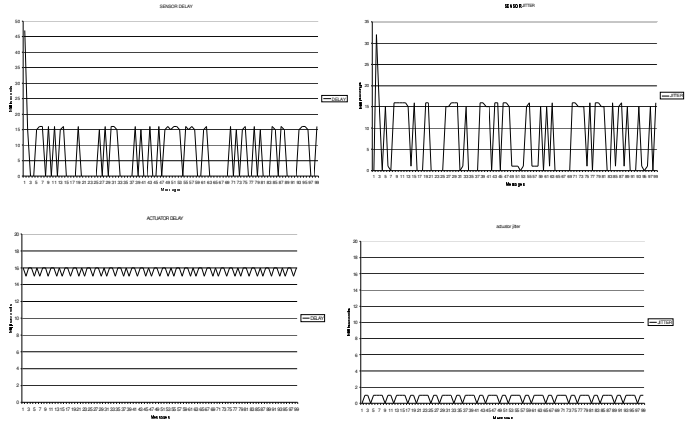

Figure 1- Test results extracts for sensors (up) and actuators (down). Delay on the left and jitter on the right.

We present here extracts from the results for delay and jitter for two of those tests, one for a simple simulated position sensor, and another for a simple force actuator.

The tests show that the combined sensor-middleware delay varies from 0 to 16 milliseconds with jitter from 0 to 16 milliseconds, while the combined actuator-middleware delay varies from 15 to 16 milliseconds with jitter from 0 to 1 millisecond. These values prove that the solution is adequate for small sensor and actuator networks around a central coordinating device, such as the ones found in mobile and pervasive augmented reality scenarios.

V. CONCLUSION

The SENSACT API is a middleware package for sensor/actuator deployment in pervasive augmented reality scenarios. The middleware allows the creation, management and communication of key components of the personal area network level of a three-level pervasive augmented reality games middleware platform. Future work in this API will address new kinds of sensors and actuators, optimisation and extension of the API, and further testing with real devices.

ACKNOWLEDGMENT

This work was partly financed by the Portuguese Foundation for Science and Technology (FCT) and by the IST FP6 E-NEXT NoE.

REFERENCES

[1] Kimmo Raatikainen, Henrik Bærbak Christensen, Tatsuo Nakajima, “Application Requirements for Middleware for Mobile and Pervasive Systems”, Mobile Computing and Communications Review, Volume 6, Number 4, October 2002, pp. 16 – 24 , ACM Press

[2] Keith Mitchell, Duncan McCaffery, George Metaxas, Joe Finney, Stefan Schmid and Andrew Scott, “Six in the City: Introducing Real Tournament – A Mobile IPv6 Based Context-Aware Multiplayer Game”, Proceedings of NetGames'03, May 22-23, 2003, Redwood City, California, USA, pp. 91-100, ACM Press

[3] Hideyuki Tamura, Hiroyuki Yamamoto, and Akihiro Katayama, “Mixed Reality:Future Dreams Seen at the Border between Real and Virtual Worlds”, Virtual Reality, November/December 2001, pp. 64 –70, IEEE

[4] Nokia – Developer resources (Forum Nokia), http://www.forum.nokia.com/, Accessed April 2004

[5] Sony Ericsson Developer World, http://developer.sonyericsson.com/, Accessed April 2004

[6] Pedro Ferreira, “Network Middleware for Large Scale Mobile and Pervasive Augmented Reality Games” in Proc. of the CoNext 2005 - ACM Conference on Emerging Network Experiment and Technology, pp. 242-243, CoNext 2005 - ACM Conference on Emerging Network Experiment and Technology, Toulouse, France, October-2005

[7] Pedro Ferreira, João Orvalho, Fernando Boavida, ”Large Scale Mobile and Pervasive Augmented Reality Games”, in Proc. of the EUROCON 2005 - The International Conference on "Computer as a Tool", pp. 1775-1778, Vol. 1, # 1, EUROCON 2005 - The International Conference on "Computer as a Tool", Belgrade, Serbia and Montenegro, November-2005

[8] João Orvalho, Pedro Ferreira and Fernando Boavida, “State Transmission Mechanisms for a Collaborative Virtual Environment Middleware Platform”, Springer-Verlag, Berlin Heidelberg New York, 2001, pp. 138-153, ISBN 3-540-42530-6 (Proceedings of the 8th International Workshop on Interactive Distributed Multimedia Systems – IDMS 2001, Lancaster, UK, September 2001)