Demo Abstract: Using Wireless Sensor Networks to Develop Pervasive Multi-player Games

Orestis Akrivopoulos, Marios Logaras, Nikos Vasilakis,
Panagiotis Kokkinos, Georgios Mylonas and Ioannis Chatzigiannakis
ichatz@cti.gr
Computer Engineering and Informatics Department, University of Patras, Greece
Research Academic Computer Technology Institute (CTI), Patras, Greece

ABSTRACT

In this work we present two mobile, locative and collaborative distributed games that are played using wireless sensor devices. We briefly present the architecture of the two games and demonstrate their capabilities. The key characteristic of these games is that players interact with each other and their surrounding environment by moving, running and gesturing as a means to perform game related actions, using sensor devices. We demonstrate our system's implementation, which uses a combination of JAVA Standard and Mobile editions.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; K.8.0 [**Computing Milieux**]: Personal Computing—*Games*

General Terms

Design, Experimentation

Keywords

multi-player games, pervasive games, mobile, SUN SPOTs, wireless sensor networks

1. INTRODUCTION

Many different applications have been proposed for wireless sensor networks. However, very few are related to mobile, interactive, multi-player games where users carry devices with sensing capabilities. Games have been a major part of the computer industry for the last decades, and are generally recognized as a means of pushing the technological boundaries, both in hardware and software. Recent advancements in mobile phones technology have produced new products that integrate various kinds of sensors into the handsets. Given that in 2008 the total number of mobile phone subscribers has well surpassed the number of 3 billion, there exists a significant user base for using such devices to play pervasive games. It is our belief that there is great potential in combining sensors and mobile devices to produce exiting entertainment applications.

Although there have been some attempts to develop multi-player games that rely on devices sensing the real world, these works are limited in number and scope. Examples of games combining WSNs with multi-player games are [3, 4, 2], ranging from virtual pet games to cooperative hidden treasure games. There are also some examples of using 3-axial accelerometer sensors to control 3D games.

In this work we present two multi-player pervasive games that are based on wireless sensor networks. Our system architecture is based on a hierarchy of layers for increased *scalability* (large numbers of users can participate in the same game) and *heterogeneity* (users may carry devices of different technologies and capabilities). Both games implement services that allow the wireless devices to localize in indoor environments, perform sensing tasks while on the move, coordinate their actions and allow delay tolerant communication. We believe that our work demonstrates the wide range of games that can be developed based on WSN.

2. OVERVIEW OF OUR SYSTEM

The architecture of our system is based on a hierarchy of layers where each layer is comprised of one or more peers (see Figure 1). Each layer is assigned a particular role in the game. Each peer may be a traditional networked processor or a wireless sensor device.

Each player carries a wireless sensing device that executes a software component (called the *player peer*). Player peers communicate with a collection of stations that form the wireless backbone infrastructure. They are used to monitor the evolution of the game and interact with nearby players. When another peer is discovered the user is prompted for further action. The user can use the sensors and buttons to explicitly trigger actions for interaction. Each action will cause the exchange of data and may involve neighboring peers. Player peers provide services that allow them to interact even when they are disconnected from the network infrastructure for extended periods of time.

The wireless backbone is established by *station peers*, each controling a specific physical area. During the initialization of the system one station peer becomes the *game engine*, responsible for the coordination of the infrastructure. The stations communicate with the users' devices either through local ad-hoc networks or via personal area non-IP networks. They provide location-aware and context-aware services to the player peers. The game engine communicates with the *world peer*, a central peer that is accessible via the Web and offers high level services. It maintains the player records, gathers player related statistics and keeps track of game history. The world peer is updated asynchronously to avoid computational and communication overhead.

SenSys'08, November 5–7, 2008, Raleigh, North Carolina, USA.
ACM 978-1-59593-990-6/08/11.

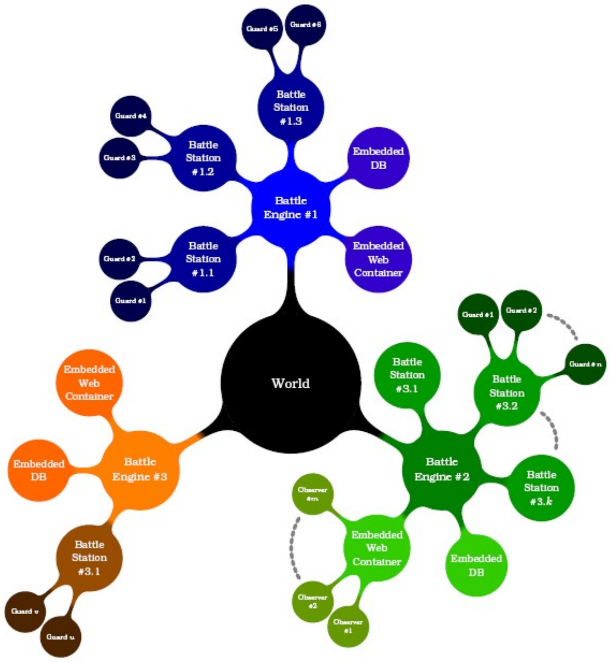

Figure 1: Overview of the game World

Our system allows the implementation of games that require near real-time response to specific aspects of the game (e.g., role playing and sports games); it offers location-oriented services for games that need players to visit specific places (e.g., puzzle, strategy and treasure-hunt games); it enables the development of services that can augment the physical reality (e.g., via Google Earth); it allows games to be played by devices that do not offer Internet connectivity.

3. IMPLEMENTATION OVERVIEW

There exist various different technologies of wireless sensor devices targeting low energy consumption, sensing modularity and reliable communication. Recently Sun Microsystems developed the SPOT platform [1] that attempts to overcome the challenges that currently inhibit development of tiny sensing devices. The device uses the IEEE 802.15.4 standard. It is a small, battery operated, device running the Squawk Java Virtual Machine without an underlying OS. Given this new hardware platform, we implemented our system using the Java programming language and the available technologies. SUN SPOTs provide a flexible hardware and software platform for developers to experiment, prototype whatever they can imagine and innovate.

4. DEMONSTRATION HIGHLIGHTS

We are going to demonstrate two example games implemented through our system. In the *Moving Monk* game the general idea is that players (*the monks*) have to find the locations of the stations (which we call *the temples*) and to perform specific actions (which we call *prayers*) inside them. A Temple is defined by the coverage range of a game station. Temples can be visited in any order and specific gestures must be used, representing actions related to the game. The first monk to visit all temples and pray is the winner of the game. In the *Assassin Apprentice* game one player is the *master* and the others are the *apprentices*. The players are not aware who is their *master*. The goal of the *master* is to locate the apprentices while hiding her role and eliminate them by performing specific gestures. The goal of the *apprentices* is to prolong their participation in the game by avoiding the *master*. If the *apprentices* uncover the *master* they can eliminate her by combining their powers, that is, perform specific gestures simultaneously.

Players can access the web interface in order to monitor their progress. Other users can be notified about the progress of the game by connecting to RSS feeds. Also users can view the physical position of the players via the well-known Google Earth application. The world peer provides KML feeds for visualizing the position of the active players.

The demonstration will feature a limited number of three stations and one engine to showcase the concepts of two games developed. There will be a total of 20 SUN SPOTs available for the attendants to use and participate in the game. A video of the game played with more stations and players will also be available for demonstration purposes.

We believe that these demonstrations will give insight to the variety and the joyfulness of the games that can be produced. These games feature location-aware services, demonstrate player interaction, offer context-aware services and support delay tolerant networking.

Acknowledgements

The games were partly implemented by students during the Spring Semester 2008 in the Department of Computer Engineering and Informatics, University of Patras, Greece, as a part of the Distributed Systems II undergraduate class. We wish to thank Dimitrios Bousis, Dionysios Efstathiou, Athanasios Gerostamoulos, Haris Koutsouridis, Andreas Loukas, Alexandros Nafas, Georgios Oikonomou, Ioannis Patlakas, Dimitrios Petrochilos, Aourela Sehou, Irini Thireou, Panagiotis Tsirigotis and Lampros Tsolakos, that also participated in the class, for useful ideas. This work has been partially supported by the European Union under contract number IST-2005-15964 (AEOLUS) and ICT-2008-215270 (FRONTS).

5. REFERENCES

[1] Sun™Small Programmable Object Technology (SUN SPOT). http://www.sunspotworld.com/.

[2] R. Ballagas, A. Kuntze, and S. P. Walz. Gaming tourism: Lessons from evaluating rexplorer, a pervasive game for tourists. In *Proc. of 6th Intl. Conference on Pervasive Computing*, pages 244–261. Springer-Verlag, 2008. Lecture Notes in Computer Science, LNCS 5013.

[3] L. Liu and H. Ma. Wireless sensor network based mobile pet game. In *NetGames '06: Proceedings of 5th ACM SIGCOMM workshop on Network and system support for games*, page 30, New York, NY, USA, 2006.

[4] S. Mount, E. Gaura, R. M. Newman, A. R. Beresford, S. R. Dolan, and M. Allen. Trove: a physical game running on an ad-hoc wireless sensor network. In *sOc-EUSAI '05: Proceedings of the 2005 joint conference on Smart objects and ambient intelligence*, pages 235–239, New York, NY, USA, 2005. ACM.