

RFID-based Distributed Shared Memory for Pervasive Games

Michel Simatic¹ and Annie Gentès²

¹ Institut Télécom, Télécom & Management SudParis, 9 rue Charles Fourier, 91011 Evry Cedex, France

Michel.Simatic@it-sudparis.eu

² Institut Télécom, Télécom ParisTech, 46 rue Barrault, 75013 Paris, France
gentes@telecom-paristech.fr

Abstract. The goal of our work is to give a user equipped with an RFID-enabled mobile handset (mobile phone, PDA, laptop. . .) the ability to access contents of distant elements of the system (tags or handsets), without physically moving to them and without using a Wireless Area Network. Our solution consists in an RFID-based Distributed Shared Memory (RDSM). After describing RDSM, we present its demonstration: a pervasive game which relies on RDSM to analyze data stored in distant elements (without any Wireless Area Network). We conclude by presenting the analysis of the players' feed back.

Keywords: Distributed memory, RFID, NFC, Vector clocks, Pervasive game.

1 Introduction

RFID tags are interesting components for pervasive games. They are inexpensive, easy to deploy and robust. Moreover a player equipped with an RFID-enabled handset can easily interact with the tags. Finally we can take advantage of Near Field Communication (NFC) technology, a subclass of RFID where a reader and a tag can communicate only if they are a few centimeters away from each other. Using NFC guarantees that the player will not be bothered by other tags while interacting with a given tag.

Our goal is to provide an architecture where a mobile handset can get the value of data stored in distant RFID/NFC tag. Classical solutions are based on the use of a server [7]. But a Wireless Area Network is required to communicate with a server. This induces installation and/or operational costs. Our goal is to do with neither a server, nor Wireless Area Network.

There are pervasive games which use RFID/NFC. But they do not meet all of our constraints. For instance, in *Save the Princess!* game, each time a player enters a room, *TinyLIME* middleware informs their terminal about the data stored in the tag located in the room [3]. The terminal displays the virtual contents associated to this room without communicating with a server. But it cannot give any information about data stored in the tag of a nearby room. On

the contrary, in *PAC-LAN* game, players are able to know what happened on a distant NFC tag [5]. Each time a tag is scanned by a player's mobile phone, the mobile sends the tag identifier to a central server. The server sends back game-related information to the mobile. And this information may concern distant tags. But, to achieve this, the game uses a server and a Wireless Area Network.

This paper presents a demonstration of our solution, the RFID-based Distributed Shared Memory (RDSM) [7]. Data are stored in a memory which is distributed among the different tags and mobile handsets (mobile phone, PDA, laptop...). Moreover, this memory is replicated on each tag and each handset of the system. Thus a handset can get the value of the data stored on a distant tag or handset. It just has to query its own replica. Whenever a mobile handset meets a tag (or another handset), their respective replica are made consistent by comparing the vector clock values coupled to the replicas.

Section 2 details RDSM. Then section 3 presents the proof of concept. Finally section 4 we present some conclusions.

2 Description of RDSM

The system we consider is made of two types of elements: RFID tags and mobile handsets (See Figure 1). Each element holds data which can only be modified by this element¹. We note DM the distributed memory made of all of these data. Each element e holds a replica DM_e of the distributed memory. As in [4], the main usage of this replica is to offer a local view of DM to e : e can make queries on the contents of any elements of the system at any time. We note $DM_e[e']$ the view element e has of the contents of DM hold by element e' . In particular, $DM_e[e]$ contains the part of DM hold by e . Each element e holds also a vector clock VC_e which is used to propagate operations done on DM .

To do so, whenever a mobile handset comes in contact with a tag (or another handset), these two elements compare vector clock values and update their own replica of DM (see [7] for algorithms). Thus, each element takes advantage of the knowledge of the other one to get more recent information concerning DM evolutions.

Usually, a vector clock element is a logical clock, incremented upon each update of its associated data [6]. In our demonstration, to save space on each tag, $VC_e[e]$ holds the timestamp of the last update or query done on $DM_e[e]$. Meanwhile, $VC_{e', e' \neq e}[e]$ holds the timestamp of the last operation done on $DM_e[e]$ which element e' is aware of.

RDSM faces staleness and scalability issues. They are discussed in [7].

3 Demonstration

The demonstration of RDSM is based on *Plug: Secrets of the museum* (PSM) pervasive game [8], developed in the context of the *PLUG* project [1]. 48 virtual

¹ In the case of data hold by a passive tag, these data can only be modified by a handset which is in contact —via RFID/NFC protocol— with the tag.

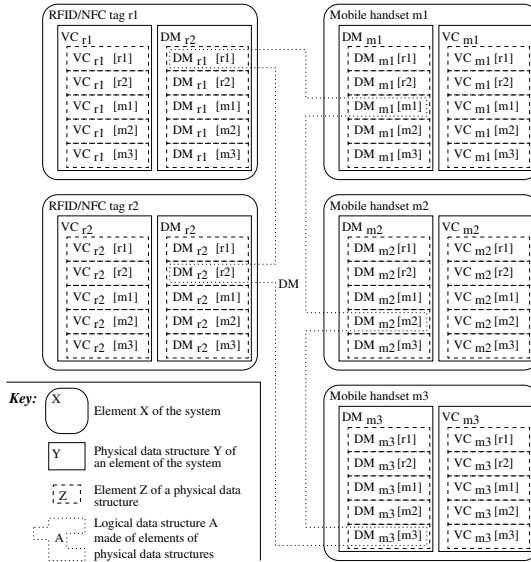


Fig. 1. Data present in a system made of 2 RFID/NFC tags and 3 mobile handsets

playing cards represent objects of a French museum called *Musée des arts et métiers*. These cards are dealt between 16 NFC tags (1 card per *Mifare* tag, each of them being equipped with 1 Kbyte of RAM) and 8 mobile phones (4 cards per Nokia 6131 NFC mobile). The players' goal is to collect cards of the same family on their mobile. Players use their mobile to swap a card with a tag or another mobile. They can also look at the hint function that indicates which tags and/or mobiles contain a card interesting for the player.

PSM uses data structure presented in section 2 as follows. DM contains the 48 cards. If e and e' are any element of the system, $DM_e[e']$ is 1 byte representing the value of the card stored. $VC_e[e']$ is 1 short (2 bytes) representing the real-time clock —formatted as the number of seconds since the beginning of the game session— of the last update or query on $VC_e[e']$ as seen by e . Thus DM_e and VC_e occupy 144 bytes in the memory of e .

The demonstration consists in having several users/players. Each one is given one of the eight mobiles to do exchanges with tags and other mobiles. This updates DM_{mobile} and VC_{mobile} . Their contents can be displayed by entering a cheat code. Thus, users can check the results of the hint function. They can also check if there are stale data in DM_{mobile} .

4 Conclusion

In this paper, we have presented an RFID-based Distributed Shared Memory (RDSM). As illustrated by the demonstration, RDSM is a good candidate for pervasive games which rely on RFID/NFC tags, must provide players hints

on data hold by distant elements of the system, with no (or limited) use of a Wireless Area Network. Such game is based on the mobility of contents and players to update the overall vision of the state of the game. Mobility and the potential for reading and writing tags create the game momentum.

The analysis of the players' feed back showed that three main features characterize mobility when it is connected to pervasiveness [2]. First, mobility appears as a way to read and collect information. Second, it is a tool to virtually mark the environment and the artifacts. Moving can be akin to "writing" a new scenario. Third, people become part of the network propagating and refreshing information not only on their mobiles but also on the RFID displays.

References

1. PLUG: PLayer Ubiquitous Games and play more (August 2009), <http://cedric.cnam.fr/PLUG/>
2. Gentès, A., Jutant, C., Guyot, A., Simatic, M.: Designing mobility: pervasiveness as the enchanting tool of mobility. In: Proceedings of the 1st international ICST Workshop on Innovative Mobile User Interactivity (IMUI 2009), San Diego, USA, ICST (October 2009)
3. Mottola, L., Murphy, A.L., Picco, G.P.: Pervasive games in a mote-enabled virtual world using tuple space middleware. In: NetGames 2006: Proceedings of 5th ACM SIGCOMM workshop on Network and system support for games, p. 29. ACM Press, New York (2006)
4. Murphy, A.L., Picco, G.: Using LIME to Support Replication for Availability in Mobile Ad Hoc Networks. In: Ciancarini, P., Wiklicky, H. (eds.) COORDINATION 2006. LNCS, vol. 4038, pp. 194–211. Springer, Heidelberg (2006)
5. Rashid, O., Bamford, W., Coulton, P., Edwards, R., Scheible, J.: PAC-LAN: mixed- reality gaming with RFID-enabled mobile phones. *Computers in Entertainment* 4(4), 4–20 (2006)
6. Saito, Y., Shapiro, M.: Optimistic replication. *ACM Comput. Surv.* 37(1), 42–81 (2005)
7. Simatic, M.: RFID-based replicated distributed memory for mobile applications. In: Proceedings of the 1st International Conference on Mobile Computing, Applications, and Services (Mobicase 2009), San Diego, USA, ICST (October 2009)
8. Simatic, M., Astic, I., Aunis, C., Gentes, A., Guyot-Mbodji, A., Jutant, C., Zaza, E.: "Plug: Secrets of the Museum": A pervasive game taking place in a museum. In: Natkin, S., Dupire, J. (eds.) Entertainment Computing – ICEC 2009. LNCS, vol. 5709, pp. 302–303. Springer, Heidelberg (2009)