A spatial computing approach for integrity checking of objects groups

Michel Banâtre
INRIA Rennes
Campus Universitaire de Beaulieu
35042 Rennes Cedex - France
Email: banatre@inria.fr

Fabien Allard
INRIA Rennes
Campus Universitaire de Beaulieu
35042 Rennes Cedex - France
Email: fabien.allard@inria.fr

Paul Couderc
INRIA Rennes
Campus Universitaire de Beaulieu
35042 Rennes Cedex - France
Email: paul.couderc@inria.fr

***Abstract*—Integrity checking is important in many activities, such as logistic, telecommunication or even day to day tasks such as checking for someone missing in a group. While the computing and telecommunication worlds commonly use digital integrity checking, many activities from the real world do not beneficiate from automatic mechanisms for ensuring integrity. We propose a spatial computing approach where groups of physical objects tagged with RFID chips are similar to network packets and group integrity can be checked at relevant places.**

I. Introduction

Integrity checking is important in many activities, both in the real world and in the information society. Essentially, it consists in verifying that a set of objects, parts, components, people remains the same along some activity or process, or remains consistent against a given property (such as a part count).

In the real world, it is a common step in logistic: objects to be transported are usually checked by the sender (for their conformance to the recipient expectation), and at arrival by the recipient. When a school get a group of children to a museum, people responsible for the children will regularly check that no one is missing. Yet another common example is to check for our personal belongings when leaving a place, to avoid lost. While important, these verification are tedious, vulnerable to human errors, and often forgotten.

Because of these vulnerabilities, problems arise: E-commerce clients sometimes receive incomplete packages, valuable and important objects (notebook computers, passports etc.) get lost in airports, planes, trains, hotels, etc. with sometimes dramatic consequences.

While there are very few automatic solutions to improve the situation in the real world, integrity checking in the computing world is a basic and widely used mechanism: magnetic and optical storage devices, network communications are all using checksums and error checking code to detect information corruption, to name a few.

The emergence of Ubiquitous computing and the rapid penetration of RFID devices enables similar integrity checking solutions to work for physical objects, using *spatial computing principles*. The purpose of this paper is to present the design of such a system, and one of its application. The paper is organized as follows: in the second section, we briefly introduce the notion of spatial computing considered here. The third section details the integrity checking problem for physical objects. Fourth section presents the design and implementation of the Ubi-Check solution. Finally, some related works and perspectives are discussed.

II. Spatial computing

Computation supported by physical processes and real world objects is not new, and various programming models relying on this concept has been proposed in the past, such as [1]–[3]. The general principle is to associate digital information to physical objects, leveraging on the physical space to support data structures (organized spatially) and mobility to support computing process. The spatial configuration of objects and their movements then implicitely control an information system. A trivial example of such a system is a shopping cart where items are tagged by their price, using RFID tags. The volume of the shopping cart implicitely reflects the total price of the shopping session, while adding or removing an item updates it.

A more complex application is for example Ubi-Bus [4]: a blind person wanting to take a given bus line carries a smart object which acts as a typed token to stop the bus: when the token generated by the pedestrian is in the area of the bus stop, the latter requests the bus to stop as it approaches, by generating a red token as shown on figure 1. It is the spatial configuration of the pedestrian, the bus stop and the bus which controls the token generation and stop requests.

Systems that relies on spatial computing principles offer two interesting properties in ubiquitous computing.

First, as they leverage on real object and existing physical processes, they are by nature pervasive and easy to use: there is usually no user interface to learn, as the system is embedded into an existing activity and existing common objects.

Second, the computing architecture is usually efficient because some of the processing is already supported by the physical processes on which the system is based on. We can see such computing architecture as being distributed over a set of physical objects. In the above example of Ubi-Bus, a centralized implementation would use a geolocalization service and the load of the system would be linear with the total number of bus and pedestrian users. A spatial implementation

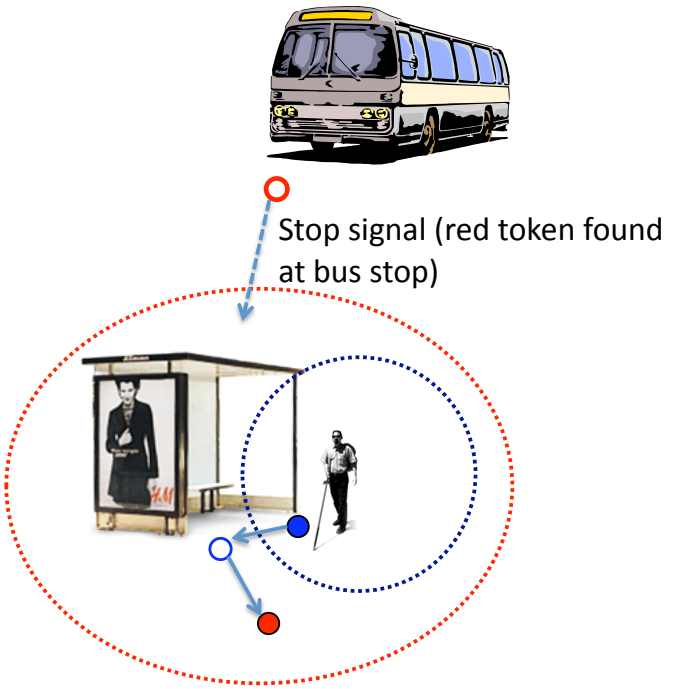

Fig. 1. Ubi-Bus: stoping a bus with a spatial computing approach

only depends on the maximum number of pedestrian around a bus stop.

In addition, considering existing physical processes as possible support for (or a part of) a computing system is an interesting perspective: solutions developped in the context of classical computing or networking can sometimes have direct applications in physical processes when revisited in the perspective of spatial computing. The system presented in this paper is a typical application of spatial computing: a well known solution for ensuring communication integrity in computer networks is revisited and applied to groups of physical objects.

Moreover, the rapid emergence of pervasive computing techologies such as RFID now enables realistic deployment of spatial computing systems.

III. Integrity checking for groups of physical objects

Consider the following application scenario, which will help identify the key issues of the problem: someone is at the airport ready to cross the security gate. He is required to wear off his jacket, his belt, to put in a container his mobile phone, his music player, to remove from his bag his notebook computer, and may be other objects... All that in hurry, with other people in the queue doing the same. Obviously, personal objects are vulnerable to get lost in this situation: objects can get stuck inside the scanner, can stack up on each other at the exit of the scanner, and it is easy to forget something while being stressed to get a flight. Another vulnerability is to get the object of someone else, such as a notebook computer of the same model.

The vulnerability is introduced because:

1) objects belonging to a common set have to be separated from each other in some occasion
2) getting them back together is not checked by reliable process

Consider what's happen in a computer network: digital objects are fragmented into "packets" which can be transported by independently of each other in the network. When they arrive at a destination point, packets are assembled together to rebuild the original object, which is checked for integrity. For this purpose, packets include additional information enabling error detection. Of course, networks are more complex than this simple view, with multiple encapsulation and fragmentation levels, but for the analogy with real objects and people, the basic principle is sufficient:

We can consider a set of physical objects as "data" which are going to be transported and eventually separated at some occasions. At some point where the set of physical object is assumed to be complete, integrity checks will take place. For instance, in our airport security gate scenario, the integrity check would be performed on leaving the zone. This approach is typical of spatial computing, as a particular area of the physical space is used to determine a condition based on the presence of physical objects.

Our goal is to propose an integrity checking system that could be integrated at strategic places to warn people when missing objects are detected, or that they are carrying someone's else object. Such a system would turn some area into smart spaces where people would not have to worry of object lost, which is interesting for trains, hotels, etc.

Such a system is only interesting if it can be realistically deployed, given the constraints of the real-world. To this end, some important requirements have to be considered :

1) ease of use and as low as possible impact on existing processes
2) low cost for the user
3) scalability and reliability
4) ease of on-site integration
5) privacy respect

The two latter requirements are very important. Integration issues can lead to death of emerging technologies or experimental systems: the cost of integrating something new into an operational infrastructure is very high, and dependence or impact on existing information systems should be as low as possible for a chance of acceptance.

Privacy concerns raise strong resistance to RFID technology [5]. As we will see, Ubi-Check addresses both these requirements thanks to its architecture based on a spatial computing principle.

IV. System design

The Ubi-Check system is based on the principle of *coupled objects*. Coupled objects are a group of physical objects that are logically associated together, meaning that they carry digital information referencing other objects from the set, or

representing their membership to the group. An important property is that this information is physically stored on the object. Typically, this information will be stored on RFID memory tags embedded on the objects.

In our application scenario, this means that users of the Ubi-Check system would have their important objects enabled with tags. Ideally, those tags could be embedded into the object at build time by the manufacturer, but user installed tags could of course be added for objects not ready for the service.

Then, there are two procedures in the system. A first one consisting of associating all the objects of a group (i.e. the objects of a person). And a second one where integrity will be checked at the appropriate places. We detail them in the following.

A. Objects association

At this step, the user presents himself in a small area in the range of an RFID writer, and a group is initialized for all its tagged objects: a signature is computed from individual identifiers of the tags. The identifiers can be those attributed at tag construction, or generated for the Ubi-Check system. The latter case is better to protect user's privacy, because new identifiers can be used each time a group is created, thereby reducing the risk of user tracking by their objects. A user could create a new group at each trip for example.

As we can see, a group is made of a set of identifiers. We need to store the group representation somewhere, such that the group integrity could be checked at appropriate places. Storage in a database is a straightforward solution, but it would require that each checkpoint could access this database, through a communication infrastructure which raises several issues:

- Deployment and operating cost
- Reliability and scalability, because checking operation would be dependant on the availability of the communication infrastructure and the remote database, and the communication load would increase linearly with the number of users of the service
- Privacy, as group representation associated with individuals would we stored in a central database. However, depending on the nature of the object identifiers and the group representation that is used, this issue may be mitigated.

These issues conflict with our design goals which motivate an alternative solution, that would not depend on a remote service to operate. In the concept of coupled objects previously mentioned, the logical association between the objects (or the group membership) is part of the physical objects themselves. This can be easily implemented with RFID tags, which in addition to the identifier part, can provide a programmable memory of up to a few kilobits. The group representation will be stored in this memory. Because the size is limited and the integrity check should be fast, the group is represented by a signature, computed by a hash code function. A good discussion of hash functions in the context of RFID is [6]. This approach enables full autonomous operation of both the association points and the checkpoints. For our application, we used `sha-256` hash functions with 96 bits identifiers for the objects and 256 bits hash values. The probability of hash collision for a group of n objects would be $(n-1)/2^{256}$ assuming perfect key distribution.

Optionnally, one particular object can be considered the *owner* of the group. The owner tag would typically be associated with an object that the user would always keep with him, such as his watch. We will see in the next section how it is used. Figure 2 sums up the association process.

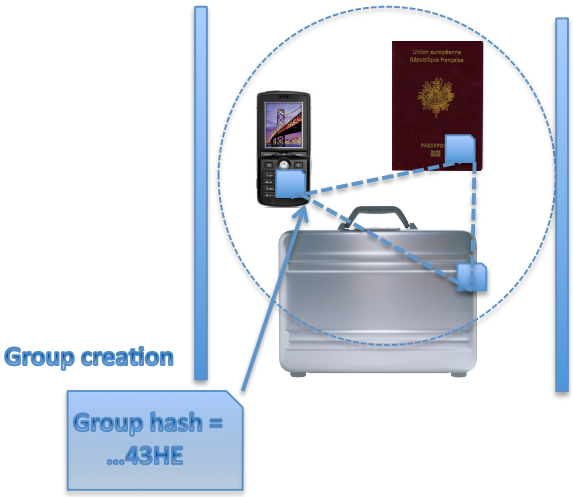

Fig. 2. Group creation

These basic principles can be extented to have objects belonging to several groups, however common RFID tags have a limited memory capacity of a few hundred of bits which limits in practice the number of group identifiers that can be enumerated. Hierachical coding allows to describe tree structures by using two parent references: a primary fingerprint (hash value) P1 corresponds to the group of the first level to which the object belongs to, while a secondary fingerprint P2 corresponds to the group of upper level to which the object belongs to. Hence, the secondary fingerprint does not exist for objects at the root of the tree. The figure 3 shows two groups O_1 and O_2, where O_1 is composed of two physical objects f_1 and f_2 while O_2 is made of of the group $O_2 = \{f_1, f_2\}$ and the physical object f_3.

B. Integrity checking

Once a group is formed, the user can move away with his objects. He can separate from his objects, but if he is passing a checking point without the complete set, a warning is shown.

The checking point is made of an RFID reader controlling a double antenna set up arranged close to each other (typically separated by one meter), in order to detect objects crossing the checkpoint. A time frame Δ_t is set to allow a group of objects to cross the gate. In practice, we have used an interval of 2 to 3 seconds with good success for typical pedestrian flow.

A cyclic buffer logs all the identifiers i passing the gate, the timestamp of the event t_i, and the signature S_i of the tag. The

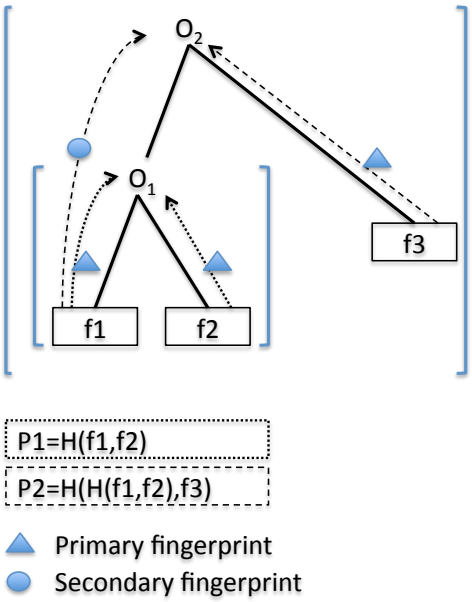

Fig. 3. Mapping hierachical structure on the tags of the objects

Fig. 4. Group creation

integrity check is triggered for each group that is reaching the end of its time frame, that is :

$\forall i$ such as $t_0 \leq ts_i \leq t_0 + \Delta_t$ and $S_i = X$

where t_0 is the timestamp when the first identifier of the group of signature X is read.

The hash code $H(i_0, ..., i_n)$ is checked against the X, and three cases have to be considered :

1) $H(i_0, ..., i_n) = X$, meaning that the set is complete. In Ubi-Check this is shown on as green status at the exit of the checkpoint.
2) $H(i_0, ..., i_n) \neq X$, and one of the identifier has the *owner status* described in previous section. This means that there is at least one missing object from the group. Ubi-Check reports a warning of missing objects. The number of missing objects is known as the owner tag includes the cardinal of the set.
3) $H(i_0, ..., i_n) \neq X$, and no identifier has owner status. Ubi-Check reports that you are carrying one or more objects that do not belong to you.

The figure 4 shows as example of a mismatch between the stored group hash and the group hash computed at the checkpoint.

C. Implementation and experimentation

A prototype of the system has been implemented: it uses FEIG HF 13.56 Mhz readers, controlled by a standard embedded PC, and standard HF read/write tags. A single unit can play both the roles of the group creator (association), and checkpoint. The figure 5 shows a simple set up with an LCD display used to report status.

Performance of the readers allow typical rate of capture of 10 tags/s, which in practice translates into checking up to 2 users per second. As we can expect with RFID technology, and

Fig. 5. A simple Ubi-Check set up

especially in this context of free mobility and on-the-fly tags acquisition, mis-reads are possible. The occurrences are highly dependant on the placement of the tags in the objects, the nature and the environment of the objects. While some advices can be suggested to the users for optimal tag placement, metal proximity and other perturbations cannot always be avoided. However, as the system is based on a multiplicity of tags, read failures typically lead to false warnings, not missed warning as failing to read all the tags from a user is very unlikely. In case of a false warning, the user can pass again the checkpoint.

The system was experimented at the “Fête de la science” in November 2008, a two days event where the general public was invited to experiment with recent scientific and technological developments. The feedback was very positive regarding the relevance of service. However, the reading reliability is currently not robust enough to allow operational deployments

in environments with sustained and continuous flow of people, such as airports. But in other contexts with relaxed constraints, such as in hotels, the system performance could be adequate.

V. Related works

RFID is a hot topic with many issues given its broad application domain and emerging success in security, accountability, tracking, etc. However, the Ubi-Check service and its underlying coupled-objects principle differs than many RFID systems where the concept of identification is central, and related to database supported information systems. In some works, the tags memory are used to store *semantic* informations, such as annotation, keywords, properties [7], [8]. Ubi-Check is in the line of this idea: RFID are used to store in a distributed way group information over a set of physical artifacts. The concept using distributed RFID infrastructure as pervasive memory storage is due to Bohn and Mattern [9].

Maintaining group membership information in order to cooperate with "friend devices" is a basic mechanism (known as *pairing* or *association*) in personal area networks (PAN) such as Bluetooth or Zigbee. Some personal security systems based on PAN for luggages were proposed [10], which enable the owner to monitor some of his belongings, such as his briefcase, and trigger an alarm when the object is out of range. A major drawback of active monitoring is the energy power which is required, as well as potential conflicts with radio regulations that can exist in some places, such as in airplanes.

Still in the context of Bluetooth, RFID has also been used to store PAN addresses in order to improve discovery and connexions establishment time [11]. It can be seen as storing "links" between physical objects, such as in Ubi-Check, but without the idea of a fragmented group. Yet another variant is *FamilyNet* [12], where RFID tags are used to provide intuitive network integration of appliances. Here, there is a notion of group membership, but it resides on information servers instead of being self-contained in the set of tags as in Ubi-Check. Probably the closest concept to Ubi-Check is *SmartBox* [13], where abstractions are proposed to determine common high level properties (such as completeness) of groups of physical artifacts using RFID infrastructures. An essential difference is that SmartBox relies on objects identification, whereas our approach relies on anonymous group information without connexion to high level semantic. Our approach is focused on group integrity checking with privacy in mind, while Smartbox is a more general approach.

VI. Conclusion

We presented an integrity checking solution for groups of physical objects based on a spatial computing principle: in strategic area of a smart space, groups of objects are checked for integrity by reading embedded digital data. The system is based on completely autonomous checkpoints, and logical group distributed over a set of physical artifacts. The strong points of this solution are its independence of any remote information system support or network support, and user's privacy respect as it is anonymous and does not rely on global identifiers. A real application to prevent personal belongings to get lost while travelling was presented. As we have seen, RF reading reliability have to be improved for some application scenarios. We have developed similar solutions for protecting bicycles (associated to their owner) using a single protected gate instead of locking each bike.

In further research we are investigating other variations of the coupled-objects concept, in particular to support objects hierarchy and to improve trust and accountability in logistic and e-commerce.

References

[1] G. P. Picco, A. L. Murphy, and G.-C. Roman, "Lime: Linda meets mobility," in *ICSE '99: Proceedings of the 21st international conference on Software engineering*. New York, NY, USA: ACM, 1999, pp. 368–377.

[2] C. Borcea, C. Intanagonwiwat, P. Kang, U. Kremer, and L. Iftode, "Spatial programming using smart messages: Design and implementation," in *ICDCS '04: Proceedings of the 24th International Conference on Distributed Computing Systems (ICDCS'04)*. Washington, DC, USA: IEEE Computer Society, 2004, pp. 690–699.

[3] J. Pauty, P. Couderc, and M. Banâtre, "Spatial programming: Using the physical world as a computing system," in *UbiPhysics workshop / Ubicomp*, 2005.

[4] M. Banâtre, P. Couderc, J. Pauty, and M. Becus, "Ubibus: Ubiquitous computing to help blind people in public transport," in *Mobile HCI*, 2004, pp. 310–314.

[5] P. Peris-Lopez, J. C. H. Castro, J. M. Estévez-Tapiador, and A. Ribagorda, "RFID systems: A survey on security threats and proposed solutions," in *PWC*, 2006, pp. 159–170.

[6] M. Feldhofer and C. Rechberger, "A case against currently used hash functions in RFID protocols," 2006, pp. 372–381. [Online]. Available: http://dx.doi.org/10.1007/11915034_61

[7] M. Banâtre, M. Becus, and P. Couderc, "Ubi-board: A smart information diffusion system," in *NEW2AN '08 / ruSMART '08: Proceedings of the 8th international conference, NEW2AN and 1st Russian Conference on Smart Spaces, ruSMART on Next Generation Teletraffic and Wired/Wireless Advanced Networking*. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 318–329.

[8] T. D. Noia, E. D. Sciascio, F. M. Donini, M. Ruta, F. Scioscia, and E. Tinelli, "Semantic-based bluetooth-RFID interaction for advanced resource discovery in pervasive contexts," *Int. J. Semantic Web Inf. Syst.*, vol. 4, no. 1, pp. 50–74, 2008.

[9] J. Bohn and F. Mattern, "Super-distributed RFID tag infrastructures," in *Proceedings of the 2nd European Symposium on Ambient Intelligence (EUSAI 2004)*, ser. Lecture Notes in Computer Science (LNCS), no. 3295. Eindhoven, The Netherlands: Springer-Verlag, Nov. 2004, pp. 1–12.

[10] R. Kraemer, "The bluetooth briefcase: Intelligent luggage for increased security," http://www-rnks.informatik.tu-cottbus.de/content/unrestricted/teachings/2004/, 2004.

[11] T. Salminen, S. Hosio, and J. Riekki, "Enhancing bluetooth connectivity with RFID," in *PERCOM '06: Proceedings of the Fourth Annual IEEE International Conference on Pervasive Computing and Communications*. Washington, DC, USA: IEEE Computer Society, 2006, pp. 36–41.

[12] W. Mackay and M. Beaudouin-Lafon, "Familynet: A tangible interface for managing intimate social networks," in *Proceedings of SOUPS'05, Symposium On Usable Privacy and Security*. ACM, jul 2005, article. [Online]. Available: http://cups.cs.cmu.edu/soups/2005/2005posters/24-mackay.pdf

[13] C. Floerkemeier, M. Lampe, and T. Schoch, "The smart box concept for ubiquitous computing environments," in *Proceedings of sOc'2003 (Smart Objects Conference)*, Grenoble, May 2003, pp. 118–121.