Pervasive Pheromone-Based Interaction with RFID Tags

MARCO MAMEI & FRANCO ZAMBONELLI
Dipartimento di Scienze e Metodi dell'Ingegneria
Università di Modena e Reggio Emilia

Despite the growing interest in pheromone-based interaction to enforce adaptive and context-aware coordination, the number of deployed systems exploiting digital pheromones to coordinate the activities of situated autonomous agents is still very limited. In this paper, we present a simple, low-cost and general-purpose implementation of a pheromone-based interaction mechanism for pervasive environments. This is realized by making use of RFID tags to store digital pheromones, and by having humans or robots spread/sense pheromones by properly writing/reading RFID tags populating the surrounding physical environment. We exemplify and evaluate the effectiveness of our approach via an application for object-tracking. This application allows robots and humans to find "forgotten-somewhere" objects by following pheromones trails associated with them. In addition, we sketch further potential applications of our approach in pervasive computing scenarios, discuss related work in the area, and identify future research directions.

Categories and Subject Descriptors: I.2.11 [Artificial Intelligence]: Multiagent Systems; C.2.4 [Computer-Communication Systems]: Distributed Systems; C.3 [Special Purpose and Application-based Systems]: Smartcards.
General Terms: Algorithms, Design, Experimentation
Additional Key Words and Phrases: Stigmergy, Pervasive Computing, RFID Tags.

---

1. INTRODUCTION

Pheromone-based interaction, adopted by social insects to coordinate their activities [8], has recently inspired a vast number of researches in pervasive and distributed computing systems [2, 3, 17, 20, 27]. In these works, autonomous and mobile application agents (whether mobile software agents, humans carrying a PDA, or autonomous robots) interact with the surrounding world and with each other by leaving and sensing artificial pheromones trails, digital analogues of chemical markers, in the environment. Pheromones, by encoding application-specific information in a distributed way and by uncoupling the activities of application agents, enable to enforce adaptive and context-aware coordination activities [19].

Despite the growing interest in pheromone-based interaction, the number of implemented systems exploiting pheromones for coordinating the activities of distributed agents situated in pervasive computing scenarios is still very limited. Most of the proposals have only been simulated [3, 17], only few of them have been concretely implemented by deploying pheromones in shared virtual data spaces [20], a few actually deploy pheromones by means of ad-hoc physical markers such as special ink or metal dust [27] (see related work section).

Inspired by these challenges, we propose a novel approach exploiting RFID technology [29, 30] to enforce pheromone-based interaction in pervasive computing scenarios. The key idea of our approach (presented in an early implementation in [16]) is to exploit RFID tags dispersed in an environment as a sort of distributed memory in which to store digital pheromones. RFID reader devices, carried by humans or by robots, could deploy pheromone trails in the environment simply by writing pheromone values in the RFID tags around. Also, they could sense such pheromone trails by simply reading pheromone values in RFID tags nearby. Clearly, such an approach is extremely low cost and not intrusive, as RFID tags will sooner or later be present in any case in any environment. Also, it can profitably complement more centralized approaches relying on network-accessible information spaces, to increase reliability in the presence of network disconnections and to enable finer accuracies in coordination.

Relying on our simple yet flexible approach, a wide range of application scenarios based on pheromone interaction can be realized, ranging from monitoring and supporting of everyday human activities [21], multi-robot coordination [27], and impromptu coordination in challenged scenarios [15]. In this paper, after having illustrated our approach, we detail and evaluate an application to easily find – by following proper RFID pheromone trails – everyday objects forgotten somewhere in our homes. Results extracted from both tests on a real implementation with RFID-reader-equipped PDAs and robots and from simulation experiments, show that our approach is very effective in enforcing context-awareness and in facilitating coordination in physical environments, though it also exhibits some limitations induced by the limited resource capabilities of RFID tags.

The remainder of this paper is organized as follows. Section 2 briefly introduces pheromone-based interaction. Section 3 describes the RFID technology and presents some basic infrastructural requirements. Section 4 details our approach to implement pervasive pheromone-based interaction. Section 5 illustrates the object tracking application example. Section 6 exploits such application to evaluate the effectiveness of our approach and to identify its limitations. Section 7 sketches several additional application scenarios that can take advantage of our approach. Section 8 discusses related work. Section 9 concludes and outlines open research directions.

2. PHEROMONE-BASED INTERACTION

Ants and other social insects interact by spreading chemical markers (i.e., pheromones) as they move in the environment, and by being directed in their actions by the perceived concentrations of pheromones. This simple mechanism of local interactions mediated by the environment, called stigmergy, enables ants to globally self-organize their collective activities in a seemingly intelligent way and to adaptively act in an unknown environment. Because such adaptive context-aware behavior is exhibited despite the very limited abilities of individuals in acquiring and cognitively processing contextual information, systems of social insects are said to be characterized by "swarm intelligence", to emphasize the difference with "individual" intelligence [3,19].

The classical example to show the power of pheromone-based interaction is ant foraging. Ants in a colony, when in search for food, leave the nest and start wandering around. When some food is found, ants start spreading a pheromone and try to get back to the nest, thus creating a trail leading to the food source. When an ant is looking for some food, it can indirectly exploit the past experience of other ants by following an existing pheromone trail to reach previously discovered food sources. This action contributes in reinforcing the pheromone trail, since the new ant will spread pheromones in its turn. To some extent, the environment becomes a sort of distributed repository of contextual information, holding the information about all the paths to the discovered food sources. The natural tendency of pheromones to evaporate if not reinforced allows the pheromone network to remain up-to-date and to adapt to changing conditions: when some ants discover a shorter path to food, longer paths tend to be abandoned and disappear; analogously, when a food source is extinguished, the corresponding pheromone trail disappears because it is no longer reinforced [3].

Despite its simplicity, pheromone-based interaction presents several features that make it suitable in a variety of distributed and pervasive applications:

1. it completely decouples agents (i.e., ants) interactions, which occur indirectly via the mediation of pheromones. This is a very desirable feature in open and dynamic scenarios where agents do not know each other in advance and can come and go at any time;
2. it naturally supports application-specific context awareness, in that pheromones provide agents with an expressive application-specific representation of their operational environment (e.g., pheromones provide a representation of the environment in terms of paths leading to food sources);
3. it naturally supports adaptation of activities, in that pheromones represent a contextual information that, when no longer updated, tend to vanish;

4. the algorithms underlying pheromone-based interaction are simple and involve local interactions only (each ant locally deposits and follows pheromones without experiencing the burden of being involved in a distributed task).

Given these features, it is not surprising that several research proposals in area as diverse as routing in networks [3], P2P computing [2, 17], robotics [20, 27], self-assembly [25], and (as in our approach) pervasive computing, incorporate and exploit pheromone-based interaction mechanisms.

3. RFID INFRASTRUCTURE

The technology of Radio Frequency Identification (RFID) is at the core of our approach for deploying digital pheromones in an environment. RFID tags are tiny wireless radio transceivers that can be attached to objects as small as a watch or a toothbrush (see also Figure 6 later on in this paper). Tags can be purchased off the shelf, cost roughly €0.20 each and, being battery-free, do not have power-exhaustion problems. Each tag is marked with a unique identifier and provided with a tiny memory (up to some Kb) allowing to store data in the form of an array of bytes.

Suitable devices, called RFID readers, can be interfaced with portable computers and can be used to access RFID tags by radio for read or write operations (i.e., to read/write specific bytes in the RFID memory). The tags respond or store data using the power scavenged from the signal coming from the RFID reader. RFID readers divide into short- and long-range depending on the distance within which they can access RFID tags: from a few centimeters (short-range readers) up to some meters (long-range readers).

So far, RFID technology has been mostly exploited as a more robust and flexible alternative to optical barcodes for automated identification of goods, as it may be required in anti-thefts systems and in logistics [4]. More recently, their potential of applicability in pervasive computing is getting recognized, and a variety of applications and infrastructures exploiting RFID technology are being proposed [6, 10, 21]. However, to the best of our knowledge, our proposal is the first one that suggests exploiting RFID tags to bring pheromones in the real world.

3.1 Scenario Assumptions

Our approach requires a scenario in which the operational environment is densely enriched with RFID tags, and in which human users and robots carry/embed some handheld computing devices provided with a RFID reader. In the next future *(i)* many household objects and furniture will be RFID-tagged before purchase and *(ii)* handheld devices provided with embedded RFID read and write capabilities will have an increasing diffusion (for instance, the Nokia 5140 phone can be already equipped with an optional RFID reader). These factors will make our assumptions become a *de facto* situation, and will make our approach directly deployable at nearly-zero costs.

We are perfectly aware that the current industrial EPC standard for RFIDs (www.epcglobalinc.org), adopted by most major retailers, considers read-only RFID tags. Also, we know that privacy concerns are pushing for solutions to deactivate RFID tags out of their original context of use (e.g., the retailer). However, we consider these two problems as temporary, and likely to disappear in the near future. In fact:

- There is a great deal of industrial activity to investigate novel kinds of writable tags, even with large storage capacity [9], to support more flexible usages in logistics.
- Research on security and privacy issues in RFID is becoming more and more active, and valid solutions to address privacy and security concerns are emerging [1, 22].

Last but not least, the novel classes of pervasive applications enabled by the flexible use of writable tags will further push for their adoption and widespread diffusion.

On the basis of these considerations, we consider that writable tags can be attached at any – even small – object, such as a pen or a cup. We generically refer to these tags as *object-tags*. They include a unique ID and additional information stored in their memories and describing some facts about the object itself (e.g., "what=cup" "who=Franco"). Other than on objects, tags can be attached to fixed locations (e.g., doors, corridors, etc.) and to unlikely-to-be-moved objects (e.g., beds, washing machines, etc.). We generically refer to these tags as *location-tags*. Other than the unique ID, also *location-tags* can contain simple information describing the location they represent (e.g., "what=office" "who=Marco").

The only actual – yet substantial – difference between *object-tags* and *location-tags* is that the former can be mobile while the latter are assumed as fixed and can act as reference points. In our approach, a specific "*Tag-Type*" bit in the RFID memory is used to distinguish between object-tags and location-tags, and 4 bytes of the memory are used to store two simple numeric key-value pairs (associating byte values to concepts according to a simple ontology [15]) describing some facts about objects and locations.

Other than the information stored within a tag itself, and provided that a connection to the Internet is available to users (e.g., WiFi), it is possible to exploit a central database to gather additional information about the RFID tags existing in the environment and about what they represent. Simply, database entries can map the tag ID to a name, a spatial location, and to any other information one may wish to store. A user, after having read a tag, can access the database via WiFi, lookup the tag ID, and retrieve additional information about the object/location corresponding to the tag. Our approach does not require the presence of WiFi Internet connection and of the database, though it can be well complemented by it.

3.2 Location and Motion Estimation Using RFID Tags

The presence of several *location-tags* in an environment enables to enforce a simple yet effective localization mechanism [7, 23]. While a user (or robot) provided with a RFID reader moves in an environment, a software agent controlling the reader unobtrusively from its user can continuously detect in-range *location-tags* to infer the current location. Specifically, the agent can simply localize the user near the last-read RFID tag. Such localization can take two forms: either the agent is provided with a map of the environment reporting the spatial coordinates of all the *location-tags*, allowing it to actually identify its coordinates in the environment; or, the agent can simply take actions on the basis of the sensed spot and report, e.g., that the user is close to the "kitchen door". In the next section, for example, we will see how the agent can follow a pheromone without knowing the global map of the environment.

Other than localization, RFID tags can be used to detect user motion. Basically a difference in the sensed *location-tags* indicates to the agent that the user is moving. More formally (see Figure 1), let $L(t)$ be the set of *location-tags* being sensed at time t. It is easy to see that an agent can infer that the user is moving when $L(t) \mathrel{!=} L(t-1)$. As we will see in the next section, this is very important to trigger pheromone propagation.

It is worth noticing that this kind of service requires a RFID reader with a rather small reading range. If, for example, the reader would be able to read tags within a 100m radius, localization would be extremely coarse. This is one of the reasons, other than cost and usability, that made us prefer passive rather than active (battery-powered and enabling long-range access) RFID tags, and short/medium-range RFID readers rather than long-range ones.

In addition to that simple localization mechanism, recent works [11] report on the possibility of adopting RFID to enable simultaneous localization and mapping by robots. This is a very interesting possibility in that it further supports the use of RFID-based infrastructures in scenarios where, other than localization, a map of the environment has to be built dynamically. As, for instance, in disaster scenarios, where the proposal has been originally applied (http://www.rescuesystem.org/robocuprescue).

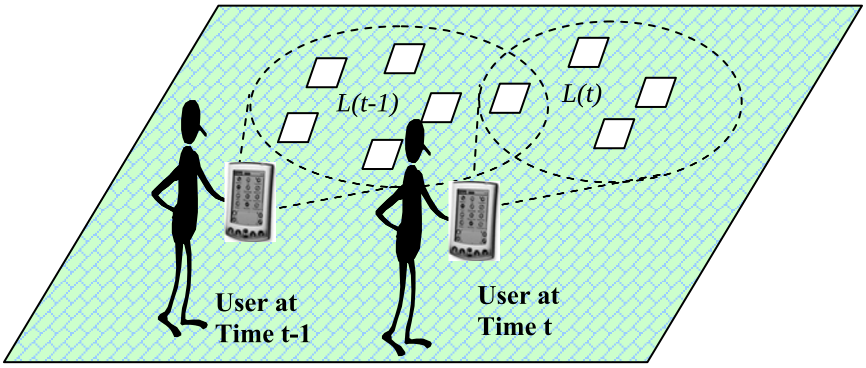


Figure 1. When the user moves, its agent gets in range with a different set of location-tags (here represented as white rectangles), and recognizes the motion.

4. DEPLOYING PHEROMONES WITH RFID TAGS

As anticipated in the introduction, pheromones are created by means of data-structures stored in RFID tags. In other words, RFID tags in the environment act as a sort of distributed environmental memory that can be used to store pheromones and to build pheromone trails.

4.1 Pheromone Deployment

The deployment of pheromones across the RFID tags distributed in an environment takes place via a software agent running on a portable computing device and in control of the associated RFID reader. Whenever instructed to start spreading a pheromone *O* (in the form of a data structure consisting in a pheromone ID and in additional information detailed in the following), the agent will write *O* in the in-range *location-tags*.

In order to spread *O* around as the location changes, the agent repeats the process by writing additional instances of such pheromone in the newly encountered *location-tags*. In particular, using the localization mechanism described in the previous section, the agent will write *O* in all the *L(t)-L(t-1)* tags as it moves across the environment. This simple process creates digital pheromone trails distributed across the *location-tags* that the agent crosses while spreading the pheromone.

Clearly, a pheromone trail consisting of only the pheromone ID is not very useful. Indeed, most applications involve agents following each other pheromone trails to reach the location where the agents that originally laid down the pheromone were directed (or, on the contrary, to reach the location where they came from). Unfortunately, an agent crossing an only-ID trail would not be able to choose in which direction to follow the trail. Thus, the data structure of each pheromone *O*, also includes a hop-counter *C(O)* associated with *O*.

More in detail, when instructed to spread a pheromone *O*, the agent initializes a *hop* counter to 0. Every time a movement is detected (*L(t)*!=*L(t-1)*), the agent reads the current value of *C(O)* in *L(t)*. If the tags belonging to *L(t)* do not have *O* or have a *C(O)* lower than *hop*, the agent stores the pheromone with value *hop*, otherwise the agent set *hop* to the value of the highest *C(O)* around. Then, it increments *hop* by 1 (see code in Figure 2). The result is a pheromone trail with an ever increasing hop counter. It is worth noticing that the fact of overwriting lower *C(O)* derives from the fact that in our applications the pheromone gradient is followed uphill. Thus, overwriting lower *C(O)* creates shortcuts in the pheromone trail.

```
hop = 0;
while(true) {
 if(L(t)!=L(t-1) {
  new = read(C(O));
  if(new == null || new < hop)
   write(C(O)=hop);
  else
   hop = new;
  hop++;
 }
}
```

Figure 2. The pheromone propagation algorithm.

The resulting overall organization of the (limited) memory of tags is shown in Figure 3. In addition to the unchangeable unique identifier, the key-value pairs, and the Tag-Type bit (used to indicate if the tag is a *location-tag*), a 7-bit "*Index*" specifies how many pheromones are currently stored in the remaining part of the memory. For each pheromone, 3 byte slots are allocated. The first byte codes the pheromone *ID*, the second codes the *C* hop counter in the first 7 bits, while the last "*Diff*" bit specifies how to propagate the pheromone (as described in the next subsection). In addition, the third byte-slot stores a timestamp, representing the time at which the pheromone has been written. This timestamp is used to support pheromone evaporation (as described in Subsection 4.4).

The adopted organization for pheromones derives from contingent choices motivated by the very limited memory of today's RFID tags and aimed at minimizing their memory occupancy. Though we do not exclude that other (more optimized) solutions are possible, we also emphasize that the evolution of the technology will soon make available tags with larger memories. Thus, without changing the basics of our approach, it will be possible to enrich pheromones with more expressive descriptions (e.g., several key-value pairs), and to add declarative rules related to their propagation patterns (e.g., to bound pheromone propagation to a specific area).

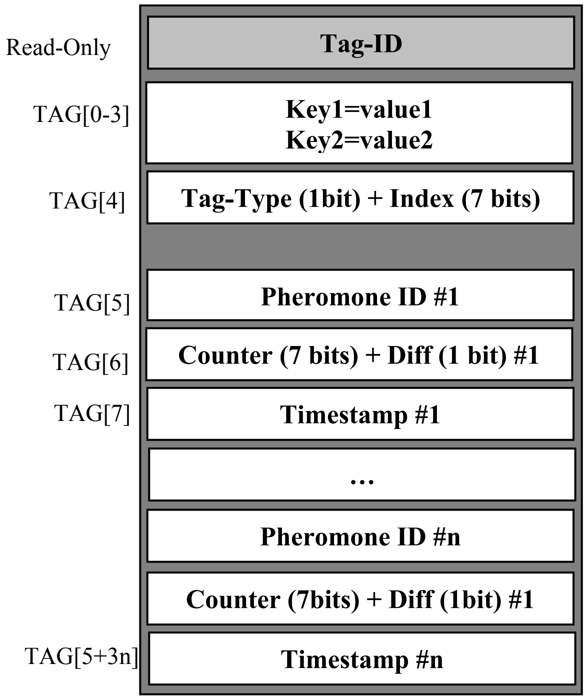


Figure 3. Memory organization of pheromones in RFID tags.

4.2 Proactive vs. Parasitic Diffusion

Unlike real-world pheromones, which diffuse around in the environment with the active support of physical laws, the passive nature of RFID tags implies that a pheromone can spread only with the support of some agent executing on some mobile computing device and controlling the RFID reader while it is moving.

The natural consequence of this fact is that a pheromone being spread by an agent does not actually diffuse around, but only in a specific direction. Once an agent starts spreading a pheromone, the resulting path of the pheromone trail will reproduce the path of the agent that originally diffused it. The problem of this mono-directional diffusion is that other agents will perceive and start following a pheromone trail only if they are lucky enough to cross it, by walking on the past steps of the agent that originally spread such pheromone. Clearly, this is not acceptable as a general solution for pheromone-based coordination.

To solve the problem, we have also integrated a form of "parasitic" diffusion of pheromones (see Figure 4). Once a pheromone trail has been diffused by an agent, other agents passing by and crossing such trail (even if having a totally different application goal) can be exploited to further support the diffusion of such a pheromone along different directions. In other words, the pheromone "parasitically" exploits the presence of a passing-by RFID reader to spread itself around along different directions (or which is the same, to branch the original mono-directional trail)[1]. When an agent diffuses a pheromone parasitically, it applies an algorithm like the one in Figure 2, but it decreases the *C(O)* value. This creates a pheromone trail that, when followed uphill, leads to the point where the original pheromone trail had been branched. Once this point has been reached, the agent can eventually follow uphill the original pheromone trail.

The "*Diff*" bit in the pheromone data structure (see also Figure 3) has the purpose of specifying whether a pheromone should be parasitically diffused or not.

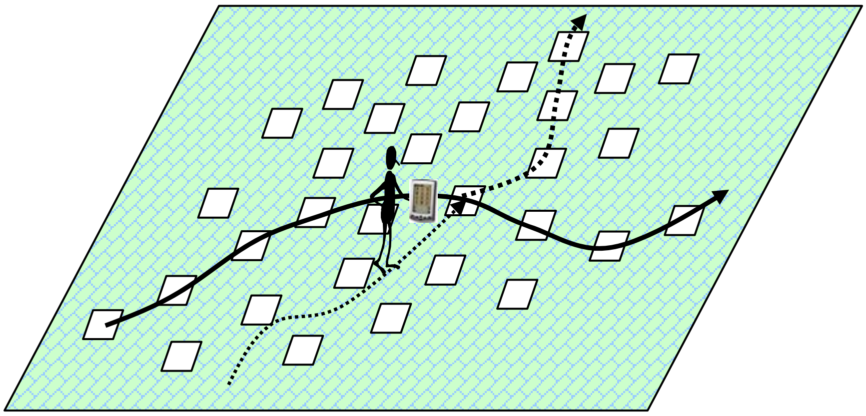


Figure 4. Parasitic Diffusion: when an agent crosses an existing pheromone trail (here represented by the chain of RFID connected via a solid arrow), the pheromone can parasitically exploit the agent to branch the pheromone trail and spread the pheromone along different directions (the branched pheromone is here represented by using dashed arrow).

4.3 Pheromone Reading and Evaporation

[1] The term "parasitic" diffusion emphasizes the fact that pheromones somewhat "theft" energy and resources from the users' PDA to prosper. However, adopting a different – more cooperative – perspective, one could consider such mechanism also as a sort of "symbiosis", where both pheromones and agents both take advantage of diffusing pheromones.

To sense pheromones, an agent trivially accesses neighbor RFID *location-tags* and reads their memories in the search for pheromones with specific IDs. Clearly, this requires that the agent knows a priori what pheromone IDs correspond to its interests, e.g., via the availability of a local table of relevant pheromone IDs. We are aware that such a solution is not very general and elegant. However, when more memory will be available in RFID tags to make it possible to associate a keyword-based descriptions to each pheromone, the recognition of the correct pheromone trail could occur without any need for a priori known tables and simply by reading the descriptions associated to pheromones (in the case of available network connectivity, RFID could also store URLs pointing to further information).

Since RFID read operations are quite unreliable, the agent actually performs a reading cycle merging the results obtained in each iteration. Given the result, the agent will decide how to act on the basis of the perceived pheromone configuration and of its own application goals. For instance, in the "object-tracking" application example described in the following, we will analyze the problem of having an agent move in the environment by following uphill a pheromone trail.

Clearly, when reading a pheromone, the agent must be ensured that it represents reasonably up-to-date situation, a problem that in natural systems is solved via a process of gradual evaporation of pheromones. In our system, the passive nature of RFID tags does not enable them to directly enforce evaporation. Therefore, we have adopted the following solution.

Each pheromone is created with an associated timestamp value *T(O)* representing the time at which the pheromone *O* has been stored (see also Figure 3). To code time into the limit of a single byte, we have adopted the solution of dividing daily time into 48 ticks (1 tick = 30 minutes). This allows us not to overflow the timestamp within 5 days of use. When pheromones are proactively (instead of parasitically) deployed, the timestamp value is set to the current time – as provided by the PDA/robot clock. This naturally represents the fact that the pheromone describes an up-to-date information. On the contrary, when the pheromones are parasitically deployed, the timestamp remains the same of the original pheromone. This is because when an agent deploys the parasitic trail, it does not have any more recent information and just re-propagates "old" data.

After reading a tag, an agent checks, for each pheromone *O* it reads, whether the associated timestamp *T(O)* is, accordingly to the agent local time, older than a certain threshold τ. If it is so, the agent deletes that pheromone from the tag. This kind of pheromone evaporation could lead to two key advantages:

1. since the data space in RFID tags is severely limited, it is useful to have a mechanism that attempts at exploiting memory at the best, e.g., by removing those pheromone trails (and freeing the associated memory slots in tags) that are there since a long time and, maybe, no longer used;
2. if an agent does not carry its personal digital assistant or if it has been switched off, it is possible that some actions will be undertaken without spreading the corresponding pheromone trails. This causes old-pheromone trails to be possibly out-of-date, and eventually corrupted.

In this context, it is fundamental to design a mechanism to reinforce relevant pheromones not to let them evaporate. With this regard, an agent spreading pheromone *O* actively, will overwrite *O*-pheromones having an older *T(O)*. From these considerations, it should be clear that the threshold τ has to be tuned for each application, because the time-frame after which pheromones are considered useless or possibly corrupted depends on the given context and application task. From the viewpoint of users/agents, the optimal threshold strongly depends on whether they search a recently moved object (in which case they would take advantage of a low evaporation threshold) or an object which did not move since a long time (in which case, they have chances of finding a pheromone trail leading to the object only if the evaporation threshold is rather high). This issue will be better analyzed in Section 6.

5. PHEROMONE-BASED OBJECT TRACKING

In this section we present a simple application we have implemented to test our approach. The application aims at facilitating the finding of everyday objects (glasses, keys, etc.) forgotten somewhere in our homes, by having objects leave virtual pheromone trails across our homes, to be easily tracked afterwards. As simple as it can be, such an application is representative of the ways that our approach can be used, and enable us to analyze further technical issues associated with our approach.

In this application (as introduced in Section 3), we assume that objects in the environment have been tagged by means of suitable *object-tags*, distinguished from the *location-tags* identifying locations in the environment.

Overall, the object tracking application works as follows (see Figure 5):

1. Users (or robots) are provided with a handheld computing device, connected to a RFID reader, and running the object tracking application.
2. The application can detect, via the RFID reader, *object-tags* carried on by the user. Exploiting the mechanism described in the previous section, an agent can spread a

pheromone identifying such objects into the available memory of near *location-tags*.

3. As the user moves, this enables to spread pheromone trails associated with the objects across the *location-tags* of the environment.
4. When looking for an object, a user can instruct the agent to read in-range *location-tags* searching the object's pheromone ID in the tags memories. This requires that the agent can locally access a table associating each object to a specific pheromone ID.
5. When the pheromone trail of the searched object is found, the user can follow it uphill to reach the object current location.

Once the object has been reached, if it starts moving with the user (i.e., the user has grabbed it), the application automatically starts spreading again the pheromone associated with the object, to keep consistency with the new object location.

This application naturally suits a multi-user scenario where a user (or a robot), looking for an object moved by another user, can suddenly cross the pheromone trail the object left before. Also, in the presence of multiple users/robots, one could effectively exploit parasitic diffusion of pheromones to increase the probability of crossing the pheromone trail being searched.

5.1 Spreading Object Pheromones

The spreading of pheromones in this application requires the agent to understand which objects are currently being carried (i.e., moved around) by its user. To perform this task unobtrusively, the agent accesses the RFID reader to detect in-range RFID tags once a second.

Let us call $O(t)$ the set of *object-tags* being sensed at time t, and recall that $L(t)$ is the set of *location-tags* being sensed at time t. If the agent senses an *object-tag* O such that $O \in O(t)$, $O \in O(t-1)$, but $L(t) != L(t-1)$, then the agent can infer that the user picked-up the object O, and that he is now moving with the object. In this situation, the agent has to spread O pheromone in the new location. To this end, the agent writes O in the available memory space of all the $L(t)$ *location-tags* that do not already contain O. This operation is performed, for every object O, upon every subsequent movement. Similarly, if the agent senses that an *object-tag* $O \in O(t-1)$, but $O \notin O(t)$, then the agent infers that the user left the object O. When this situation is detected the agent stops spreading the O pheromone. These operations create pheromone trails of the object being moved around.

It is worth noticing that the presented algorithm works best for rather short range readers where, if the above conditions are met, the object has been truly picked up. In the case of long range readers, it can happen that the areas covered by two subsequent readings overlap. If an object lies in the intersection of two of these areas, the algorithm infers that the object has been picked up, while it was not touched. This creates a spurious object pheromone near the object itself. However, for the sake of the object tracking application, this is not a big problem, in that the pheromone is propagated only in the close proximity to where the object is actually located.

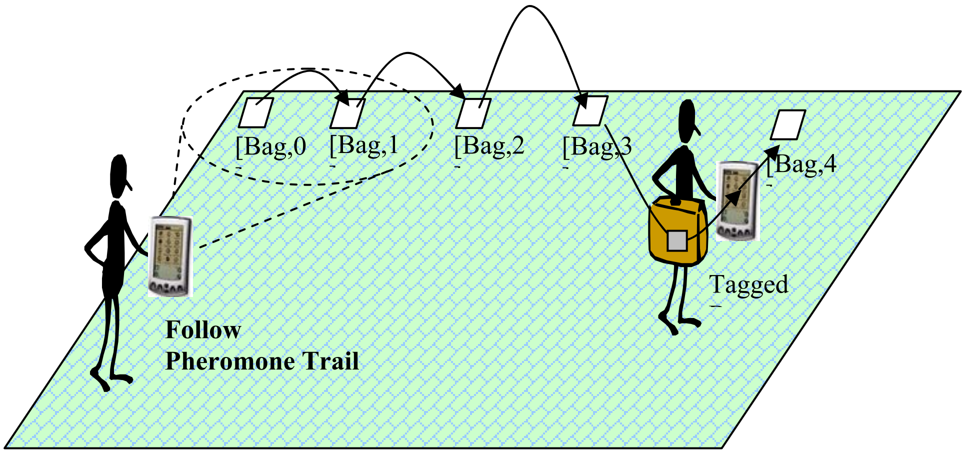


Figure 5. Object-tracking. As the user with the bag moves around, the agent on the PDA recognizes the user is carrying a tagged bag, and then spreads the "Bag"-pheromone (and the associated hop counter) in the location-tags nearby. Meanwhile or later on, another user can look for the bag by following the "Bag" pheromone trail.

5.2 Tracking Objects

Once requested to track an object *O*, the agent will start reading once per second nearby *location-tags* looking for an *O*-pheromone within the sensed *location-tags L(t)*. If such a pheromone is found, this implies that the user has crossed a suitable pheromone trail. Then, this trail has to be followed uphill (in the direction of increasing *C(O)*) to find the object. To this end, two alternatives arise: either *L(t)* contains only one *location-tag* (as it may happen with short-range RFID readers) or *L(t)* contains at least two *location-tags* having *O*-pheromones with different *C(O)* (as it may happens with medium- and long-range RFID readers).

In the former (unlucky) case, the application notifies the user about the fact he has crossed a pheromone trail, but nothing else. In such situation, the user has to move in the neighborhood, trying to find higher *C(O)* indicating the right direction to be followed (this is like dowsing – i.e., finding underground water with a forked stick – but it works!). We refer to this as *local-search.*

In the latter (lucky) case, the agent notifies the user about the fact that he has crossed a pheromone trail, and it suggests to move towards those *location-tags* having the higher *C(O)*. In the following, we will refer to this as *grad-search*, since it is like following a gradient uphill. With this regard, it is important to emphasize that *grad-search* is likely to be available only with RFID readers with a range long enough to include in *L(t)* at least two tags storing the pheromone trail. Moreover, since we do not require the presence of additional localization devices that map objects to physical coordinates, the agent suggests the user to get closer to the location having higher *C(O)* by naming it, e.g., walk to the "front door". This implies that the user has to know how to get there without further help. For robots, this implies that robots have to localize and internally code a map of the locations in the building.

In both cases, by following the agent advices, the user following the pheromone trail gets closer and closer to the object, until it reaches it.

6. EXPERIMENTS

To assess the validity of our approach and the effectiveness of the object tracking application, we performed a number of experiments, both adopting a real implementation and a simulation framework. The approach consists in testing the feasibility and usability of the system on the real implementation, and then in developing a simulation framework matching the real data in large-scale scenarios.

6.1 Real Implementation Set-Up

The real implementation consists in tagging places and objects within our department (Figure 6a). Overall, we have tagged 100 locations within the building (doors, hallways, corridors, desks, etc.) and several objects (books, laptops, cd-cases, etc.) within. Locations have been tagged with ISO15693 RFID tags, each with a storage capacity of 512 bits (each tag contains 30 writable byte slots, and can store 6 pheromones overall). Objects have been tagged with ISO14443B RFID tags, each with a storage capacity of 176 bits (each tag contains only the object ID and two key-value pairs).

Users are provided with HP IPAQ 36xx PDAs, each running Familiar Linux 0.72, J2ME (CVM – Personal Profile), and having installed a WLAN card and an Inside M21xH medium-range RFID reader, making it possible to read tags at a distance up to 20 cm (see Figure 6b).

To test the effectiveness of our approach, we have conducted several experiments organized as follows. One user carries on an object, hides it within the department and deploys a corresponding pheromone trail. Later on, two users try to find the object. One of them without any support or suggestion, another trying to identify and follow the pheromone trail associated to the object. What we have found is that, in the majority of the cases, the user following the pheromone trail is able to find the object faster. However, we have noticed that the process of following the pheromone is notably slowed down because of the rather limited range of the adopted RFID reader, implying the user to adopt a *local-search* (i.e., wandering around the trail to identify the uphill direction).

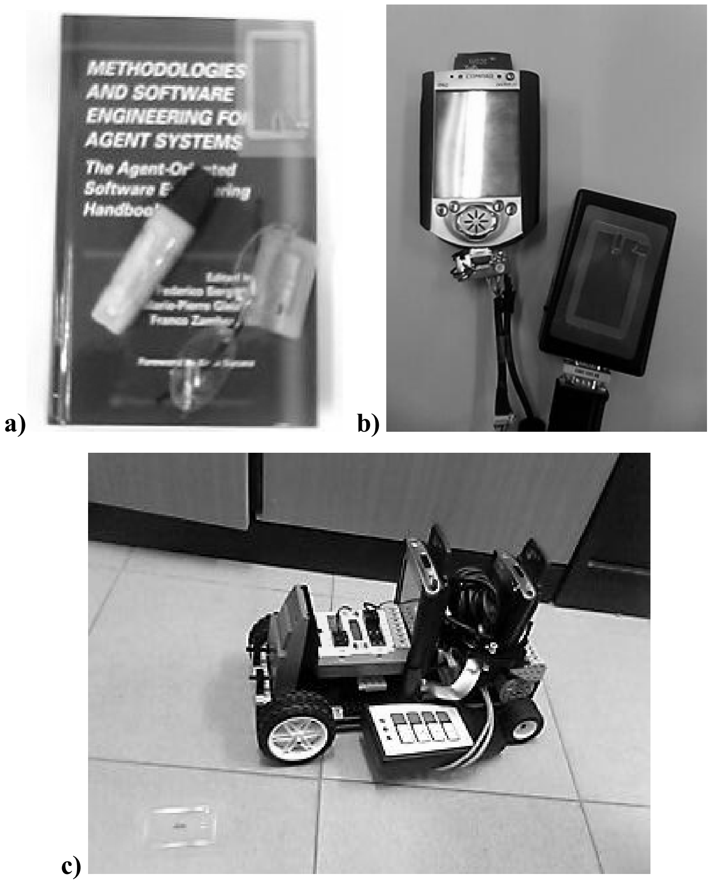


Figure 6. (a) Some tagged objects. (b) The test-bed PDA hardware. (c) The Lego Mindstorms robot with two PDAs and an RFID reader mounted onboard.

In addition, to test the effectiveness of our approach for robots other than for humans, we have developed a robot equipped with a RFID reader. In particular, we have mounted two PDAs onboard of a Lego Mindstorms robot (*www.legomindstorms.com*). One PDA is connected to an RFID reader. The other PDA is connected to the robot microprocessor[2]. Two agent applications, running on the PDAs and talking with each other via WiFi, coordinate the robot activities (Figure 6c). 100 (10x10) RFID tags have been attached to the pavement grid, and robots try to find a specific tag ID in the grid. We have then compared the behavior of a robot exploring the grid blindly (without pheromone support) to that of a robot following the previously spread pheromone trails. Also in this case, the limited range of RFID readers makes *local-search* the only possible solution. What we have found here (in contrast with the experiment performed with human users), is that the robot following the pheromone trails is not able to significantly outperform the robot exploring the grid blindly.

On the basis of these contrasting experiments, and to better unfold the effectiveness of our systems and the effects of the various parameters involved, we have also set up a set of simulation experiments.

6.2 Simulation Set-Up

To test more extensively and on a larger scale, we have implemented a JAVA-based simulation of the above scenario. The simulation is based on a random graph of places (each associated to a *location-tag*), and on a number of objects (each associated to an *object-tag*) randomly deployed in the locations-graph. Each tag is simply simulated by an array of integer values.

A number of simulated agents wander randomly across the locations-graph, collecting objects, releasing objects, and spreading pheromones accordingly. At the same time, other simulated agents look for objects in the environment eventually exploiting pheromone trails previously laid down by other agents.

The simulator allows performing a number of experiments by varying a number of parameters such as the graph size, the number of objects, the number of agents involved, the storage capacity of the tags, etc. For the sake of comparison, we have tested both the *local-search* algorithm in which the agents perceive the pheromones in their current node but cannot see the direction in which the pheromones increase, and the *grad-search*

[2] Such peculiar design is motivated by the fact that both the RFID reader and the Lego microprocessor require a private serial connection to communicate with the PDA, but the IPAQs we employed are provided with just one serial interface.

algorithm, in which the agents perceive pheromones together with the directions in which they increase. These algorithms have also been compared with a *blind-search* algorithm, in which agents explore the environment systematically, fully disregarding pheromones.

6.3 Results of the Simulation Experiments

A first group of experiments (Figure 7) aims at verifying the general effectiveness of our approach and of the object-tracking application.

We report results from two different simulation scenarios: the first consists in 100 tagged places with 100 objects (Figure 7-a); the second consists in 2500 tagged places with 500 objects (Figure 7-b). A number of 10 agents are simulated to populate these environments wandering around moving objects and spreading pheromones (only proactively – the effects of parasitic diffusion will be discussed later on) and, at the same time, looking for specific objects. In the experiments, we report the number of places visited (i.e., the number of *location-tags* perceived) before finding specific objects, for different search methods, plotted over a virtual time. In these experiments the number of objects to be tracked (i.e., the number of pheromone trails) is much lower than the tag storage capacity. Thus, tag saturation is not an issue and pheromone evaporation is unnecessary. The reported results are averaged of about 300 simulations.

Starting from a scenario free of pheromones (time zero in Figure 7-a), the more time passes the more pheromone trails get deployed by agents. *Blind-search* does not take advantage of pheromone trails: objects are found after visiting on the average half of the places. *Grad-search* takes a great advantage of pheromones: after an initial period, and when several pheromone trails have been deployed, *grad-search* starts becoming very effective: less than 10% of the places need to be visited before finding the object. *Local-search*, instead, appears not to take any relevant advantage of pheromones. This is due to the cost of orienting in the environment to find the proper direction, at least in the small-scale scenario of Figure 7-a. These results are perfectly in line with the experiments performed on the Lego robot that, by adopting *local-search* in a grid of 100 tags, were not able to significantly take advantage of pheromones. The fact that, instead, humans can take advantage of pheromones even with *local-search* derives from the fact the actual topology of a building (e.g., our department) is generally more constrained than a random graph or a grid. Thus, humans do not require repeating the process of finding the uphill directions at each and every step.

In any case, such a situation changes when getting to larger-scale scenarios (as in Figure 7-b). There, both *local-search* and *grad-search* appear reasonably effective. The performance improvements of *local-search* in this case are due to the fact that the cost of "orienting" in a local neighborhood becomes negligible when the environment is large. Thus, although the *grad-search* algorithm is always preferable, in large-scale scenarios our approach is effective even when using short-range RFID readers enabling to enforce the *local-search* algorithm only.

It is worth noticing that the standard deviation of the experiments' results tends to be quite large (i.e., about 50-60% of the average). This is due to the inherent random nature of the search process: sometimes an agent may find a pheromone trail or the object itself immediately; sometimes it may take a while. Still, for example with reference to Figure 7-a, more than 95% of the *grad-searches* perform better than the corresponding *blind-* and *local-searches*.

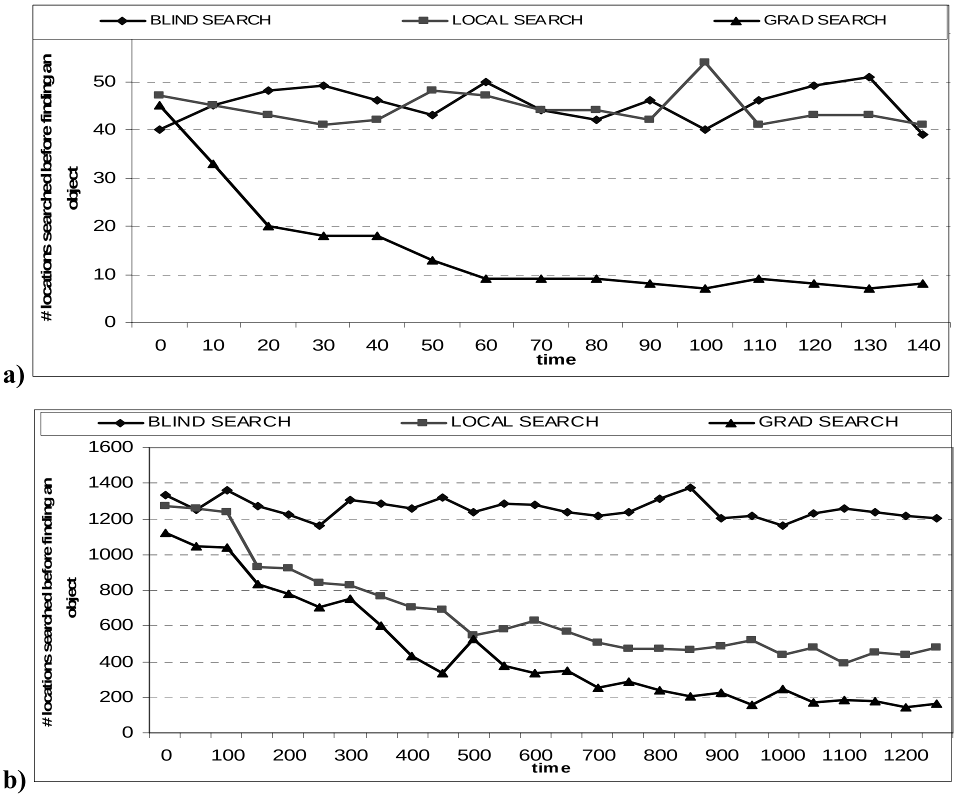


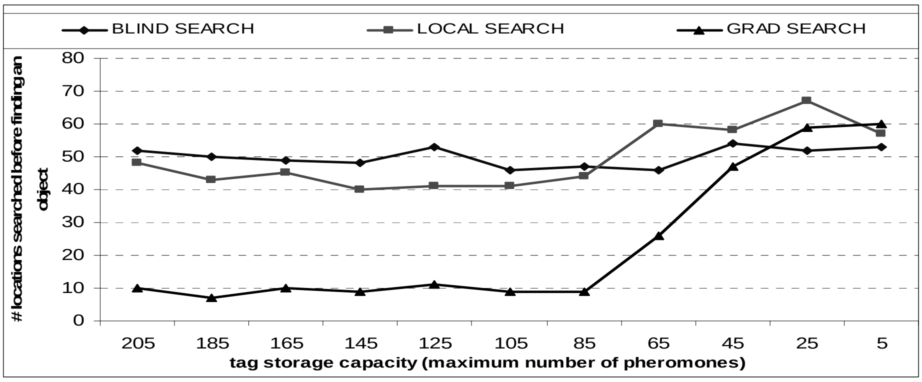
Figure 7. Number of places visited before finding a specific object plotted over time. (a) 100 tagged places. (b) 2500 tagged places.

A second group of experiments aims at exploring the effects of RFID tag storage saturation upon pheromone spread (Figure 8). This may represent a big problem: it can happen that pheromone trails can be interrupted, because there is not available space left on neighbor *location-tags*, while the object to be tracked moves away. This creates a broken pheromone trail leading to a place that is not the actual location of the object.

In Figure 8-a, we report an experiment conducted in the 100-tagged-places-environment described before. We plot the number of places visited before finding specific objects for different search methods, over a shrinking tag storage capacity. In these experiments, agents move carrying objects (and thus spreading pheromones) for 150 time steps, then they start looking for objects without picking up any of them (and thus without spreading pheromones anymore).

Let us focus on the *grad-search* method that is the most interesting in this context. It can be noticed that, when the tag storage capacity is high, we have good performance. When the capacity fall below 85 pheromones (that is – when the tag has a capacity of less than 85 * 3 bytes = 255 bytes), performance starts decaying really fast. When the capacity is lower than 25 (75 bytes), *grad-search* works equal to *blind-search*. This phenomenon is rather easy to explain: when the tag capacity is low compared to the number of objects to be tracked (i.e., the number of pheromone trails to be spread in the environment), there are a lot of broken pheromone paths degrading the performances. An agent, reaching the end of a broken pheromone trail, has no choice but starting the search from the beginning. It is important to understand that the ratio between the objects to be tracked and the tag storage capacity is the fundamental measure governing *grad-search* performance. The more the objects to be tracked, the more the knee in the *grad-search* graph moves left, because more RFID tags saturate.

Figure 8-b shows the same problem from the time perspective, with the tag capacity fixed to 50 pheromones (150 bytes). The experiment shows the number of places visited before finding specific objects, for different search methods, over time. Let us focus again on the *grad-search* behavior. It is easy to see that, when time is close to zero, *grad-search* works equal to *blind-search*, since no pheromone trails have been laid down. After some time, *grad-search* works considerably better than *blind-search*, since pheromone trails drive agents. However, as time passes, tags capacity tends to saturate: the objects are moved, but no pheromone trails can be deployed. This situation rapidly trashes performance leading back to *blind-search* performance. For instance, in our real implementation (tags with a 512 bits capacity, i.e., 20 pheromones), the above problem leads to a large number of broken trails as soon as more than 20-30 objects are being tracked.

a) 

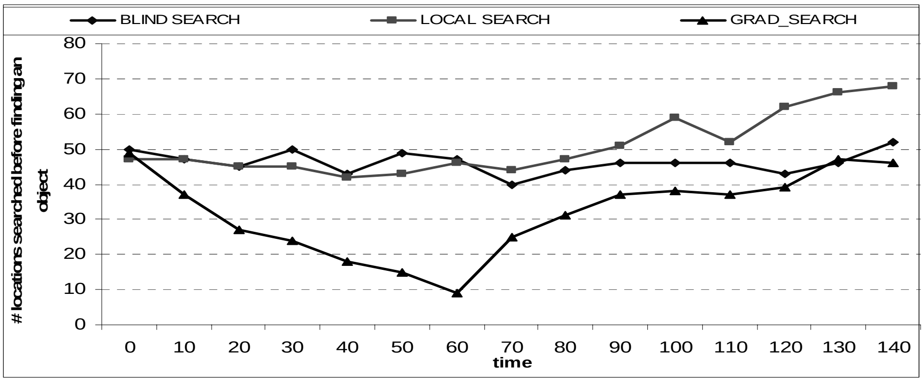
b) 

Figure 8. (a) Number of places visited before finding a specific object, plotted over shrinking tag storage. (b) Number of visited places before finding a specific object, plotted over time, when tags tend to saturate.

Finally, in the experiments of Figures 9, 10 and 11, we tried to assess whether the pheromone evaporation mechanism can help in such a situation. The experiment in Figure 9 plots the number overflows in write operations over a shrinking tag space for different pheromone thresholds τ. This number is critical because, when overflows happen, objects are moved without laying down the corresponding pheromone trail, and all the pheromones deposited up to that point break down. The experiment shows that pheromone evaporation is effective in deleting old and possibly corrupted pheromones, and when the threshold τ is low enough (pheromones become obsolete quickly enough) the number of overflows is greatly reduced, indicating that RFID tags are freed readily.

Figure 10 plots the number of places visited before finding specific objects over a shrinking tag storage capacity (the same of Figure 8-a). This time, however, only *grad-searches* are depicted and each plot is associated to a different threshold τ of pheromone evaporation. Pheromone evaporation is apparently rather ineffective: the number of locations being searched by agents before finding an object is almost independent of the evaporation threshold. This is a consequence of the considerations we have anticipated in Section 4.3. The usefulness of evaporation strongly depends on whether agents mostly search recently moved objects (in which case a low evaporation threshold is better) or objects which are not moved since a long time (in which case, a high threshold would be better not to delete old – yet relevant – pheromones). In the experiments of Figure 10, the objects to be searched are selected randomly, thus a high-threshold is good when the selected object is not moved for a long time, but it is bad if the selected object is moved recently (since a high threshold increases the probability of a write overflow). On average, changing the threshold does not improve search results.

By disaggregating the data of these experiments, as from Figure 11, the different impact of pheromone evaporation on different agents may be properly analyzed. There, the evaporation threshold τ is fixed, and data series distinguishes between those agents that mostly search objects that have not moved recently (diamond data series), those that only search objects that moved in the last 20 time steps (triangle data series), and those that search for both old and newly moved objects. It is easy to see that searches for recently-moved objects take a notable advantage from the low threshold, while searches for objects rarely moved are penalized by it.

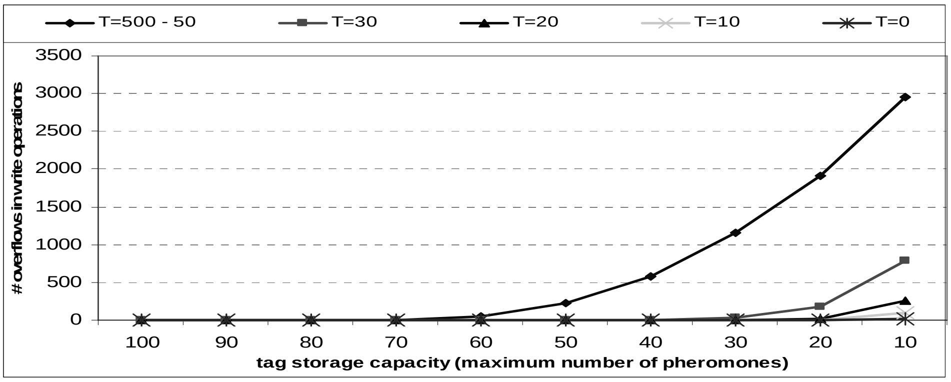


Figure 9. Number of overflows in write operations, for different evaporation thresholds τ. Thresholds between 500 and 50 never deletes pheromones (τ is too high).

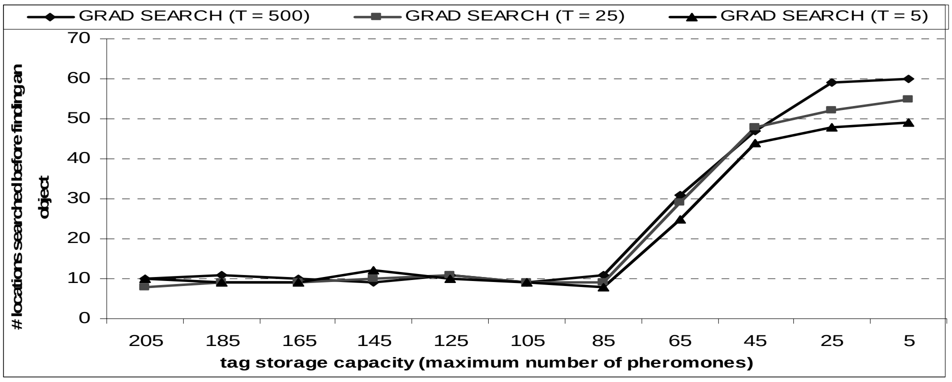


Figure 10. Number of visited places before finding a specific object plotted over a shrinking tag storage space, for different evaporation thresholds τ.

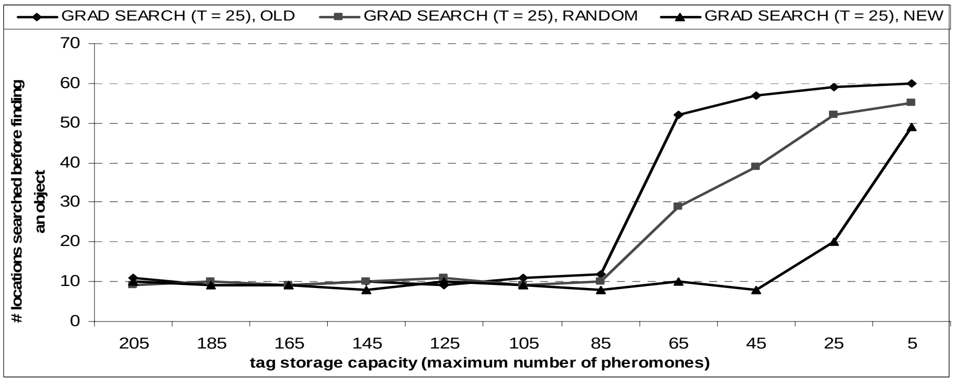


Figure 11. Number of visited places before finding a specific object plotted over a shrinking tag storage space. Objects being searched are rarely moved objects (diamond), frequently moved objects (triangle), random objects (square).

6.4 Effects of Parasitic Diffusion

When pheromones are allowed to diffuse not only in a proactive way (as in the experiments above) but also parasitically, two effects arise:

- The number of visited places before a pheromone trail is encountered decreases, which contributes scaling down the time needed to find objects, both for *local-search* and for *grad-search*. Again, this effect is more relevant in large-scale scenarios.
- The effect of memory saturation worsens, due to the increased number of pheromone trails that have to be stored in the environment.

Clearly, the above two effects are increasingly evident the more agents exist in the environment that parasitically diffuse pheromones.

6.5 Discussion

The above experiments show that our approach is effective but it also exhibits problems. From the positive side, our approach is effective in deploying pheromone trails in the environment. This can be used to enable stigmergic coordination and to achieve context-awareness. As the experiments in the object-tracking application show, such pheromone trails can be effectively exploited by agents whether they exploit *local-search* (requiring cheap and small short-range RFID readers) or *grad-search* (requiring larger RFID readers), though the latter remains preferable.

From the negative side, the main problem of our approach relates to the limited storage capacity of the RFID tags. Basically, if the number of objects to be tracked (or, more in general, if the number of pheromone trails to be deployed in an environment) is greater than the available slots on the RFID tag, in the long run the problem of tag saturation is unavoidable. Sooner or later, a new object will cross an already saturated tag, breaking the pheromone trail. The pheromone evaporation mechanism that we have implemented may help in this situation only if it is known *a priori* whether agents will more likely search for new pheromone trails only or also for old pheromones trails. In our opinion, such an assumption cannot always be granted, and it is something that a general infrastructural approach should not require.

We still do not have a general solution for this problem. Our research with regard to this topic is leading in two main directions: *(i)* we are currently researching more advanced pheromone evaporation mechanisms; *(ii)* we are considering the idea of spreading pheromone trails not only in *location-tags* but also on *object-tags*. The advantage of the latter solution would be that the more the objects in the system, the more the storage space is available for pheromones, making the system inherently scalable. The problem would be in managing the movements of *object-tags* storing pheromones, which would break the pheromone trail structure. As a partial relief from the pheromone saturation problem, it is worth reporting that (as already anticipated) recent RFID tags have a storage capacity in the order of several KB, and the trends indicate that such capacity will further increase in the near future. This will make it possible to track hundreds of objects (or, more in general, to spread a large number of pheromone trails in an environment) without suffering from tag saturation.

7. OTHER APPLICATION SCENARIOS

Pheromone interaction and stigmergy have attracted more and more researches due to their power in supporting context-awareness and adaptive coordination in a variety of scenarios. Thus, it is not surprising that even our proposal for RFID pheromone deployment could find a number of additional applications, beside the presented object-tracking application.

The value and novelty of RFID-based infrastructures can be best perceived in the context of challenged scenarios where infrastructures based on network connectivity may not be available [15, 28]. There, RFID tags provide a uniquely-cheap, easily-deployable, and likely-to-be-already-there memory infrastructure.

One could generally think of exploiting RFID pheromones to enable a group of users and robots to coordinate their movements in an unknown and challenged environment on-the-fly, without any advanced planning. As a practical example, consider an emergency rescue team (whether human, robotic, or mixed) arriving in a disaster area where no computing/network infrastructure is available, other than the (nearly unbreakable) RFID tags around. One the one hand, if the team members exploit these tags to spread pheromones around as they walk, and are instructed to stay away from existing pheromone trails, then one can have reasonable guarantees that the whole environment is explored in a comprehensive and effective way by the group [27]. On the other hand, whenever a member of the rescue team discovers something important that should be found by other members of the team (e.g., a robot finds an injured person requiring medical assistance), it can start spreading a pheromone leading to that something, so that other members (e.g., first-aid doctors passing by) can notice it.

In this context, it is important to remark that our approach appears to require the presence of RFID tags before pheromones can be spread. Although RFID tags are likely to be soon densely present in everywhere, one cannot rely on this assumption for sensible situations such as that of a rescue team in action. Still, it is possible to conceive the possibility for users or robots to physically deploy RFID tags on-the-fly, while exploring the environment, to be used for subsequent coordination. Furthermore, the future development of plastic (and printable) RFID technology [5] let us envision the possibility of enriching an RFID reader with a RFID printer, to dynamically print RFID tags in pavements, walls, or any type of surface, whenever needed.

In line with these ideas is the concept of pervasive workflow management for manufacturing systems. Standard workflow management systems are rooted on a software engine keeping track of the status of the workflow being carried on. Workers notify to this engine the tasks being completed and the engine in turn notifies the subsequent tasks that have to be carried on. RFID tags and pheromone-based interaction could complement such kinds of systems allowing to store information about the workflow status directly on the artifacts being processed. This kind of approach could lead to more situated and adaptive scenarios, where the correct manufacturing workflow could be enacted in a fully distributed way, also when the network is temporarily not available.

More in general, RFID tags can be used to help users (as well as robots) in getting aware of what's in the environment more than their natural and artificial senses can do, i.e., by reading additional information that can be provided by tags. To some extent, RFID tags may enable a sixth "digital" sense with which to gather digital information from locations and objects. While this can be simply an add-on for people with normal abilities, the additional sensing capabilities provided by RFID tags may be dramatically important for people with limited abilities, e.g., for helping visually impaired in getting aware of what's around [12]. Specifically, the use of pheromone trails can support a guided navigation toward specific locations/objects in the environment. Consider the case of a visually impaired person having mounted a short-range RFID reader on its white stick, to sense pheromone trails stored on RFID tags attached to the pavement. In this case, the ability of accessing information stored directly in the environment is dramatically important and useful, and while such information can be fruitfully complemented by additional information stored in some databases and accessed via WiFi, it can hardly be fully replaced by it.

Other than for navigation purposes, the activities of accessing (or simply getting in range) with some RFID tags can be used to achieve awareness of the activities occurring in the environment. One of the most interesting works in this direction has been presented in [21]: a software application is able to infer the users' daily activities on the basis of the objects he touches (e.g., if the user touches a teapot and a cup, the application can infer that he is preparing tea). All these facets of context-awareness – which mostly exploit information assumed to be already stored in tags, can be enriched by the ideas presented in this paper, suggesting to: *(i)* exploit RFID tags in the environment as a sort of distributed shared memory storing the history of locally occurred events; *(ii)* exploit pheromones to keep a traceable distributed track of past environmental activities. For instance, in the application for inferring daily activities, one could think that once the application recognizes that the teapot is brought to the fridge for cooling, the teapot spreads a pheromone trail leading to the fridge and indicating that some tea has been prepared and is there to cool down.

In summary, all these application scenarios show that the use of RFID tags as a distributed memory infrastructure could be valuable in several cases:

1. In challenged scenarios where no network infrastructure is available to access remote data and services, RFID could be the only viable option to share information quickly and without the need to deploy costly pervasive infrastructures such as wireless sensor networks.
2. In scenarios where the network can be available, RFID could in any case work as a robust backup infrastructure to store replicas of critical data. In the case of network glitches, applications could try to access the information stored in the RFID tags on a best-effort basis. This would enable a more autonomic system behavior capable of dealing gracefully with temporary connectivity problems.
3. Even assuming scenarios in which the network is always and reliably available, storing information in RFID tags can be useful both to enable finer and more natural means of interaction with the physical world (information is right on the objects/places it relates to, and it is implicitly accessed on a location-dependent basis) and to improve the scalability of the system (information that is meaningful and useful only in place, need not to be forwarded to some centralized storage system and need not to be searched on large data sets).

These considerations foster the ideas presented in this paper beyond the simple object tracking application.

8. RELATED WORK

In the last few years, a lot of distributed applications inspired by pheromone interaction have been proposed. However, only a few of them define actual solutions to deploy pheromone-coordinated systems in pervasive and mobile computing scenarios.

In the absence of a physical infrastructure on which to deploy pheromones, a possible solution is to provide a virtual representation of the environment and of the pheromone trails. For instance, pheromone-based approaches to coordinate unmanned airspace vehicles (UAVs) [20] and automated guided vehicles (AGVs) [32] have been implemented and tested. There, pheromones are spread in a virtual environment accessed as a sort of distributed shared memory by all UAVs/AGVs. As already explained, the novelty of our proposal is to be functional also in situations where the network connectivity is absent. In any case, nothing prevents from combining our approach with those using a virtual representation of the pheromone landscape. In particular, the two approaches could be combined so as to use RFID when the network is not available, and suitable pheromone servers when the network is present. Such kind of mixed approach would go in the direction of making the system more autonomic, i.e., capable of self-adapting to changing environments.

Another approach discussed in the literature considers implementing pheromones as an overlay data structure, realized by exploiting as an infrastructure the same agents to be coordinated. For instance, some proposals for swarm robotics [14] suggest spreading and diffusing pheromones (e.g., pheromone trails) in an ad-hoc way over the wireless network constituted by the robots to be coordinated. Such a solution presents problems related to the cost of individual robots, and to the number of robots which are required to provide a good and dense-enough coverage of the environment. Also, the approach suffers from the fact that when the ad-hoc network of robots gets partitioned, pheromone trails automatically break down. For these reasons, the solution appears most suitable for coordinating activities in modular robots and self-assembly systems, where the coverage and density of the network are automatically ensured by the fact that components are in direct contact with each other [25]. In any case, a suitable combination of the proposed RFID approach and of overlay-based approaches could provide improved performance. For example, the RFID tags could be used as a cache storing pheromone data. Should the overlay network get partitioned, some backup data could be retrieved from the RFID tags. This would let the system to survive connectivity problems.

If one could assume the presence of an active pervasive network infrastructure, i.e., a wireless sensor network deployed in the environment, then one could effectively use the nodes of such a network to store and access pheromones, as described by Li and colleagues [13]. This approach is somewhat similar to our RFID implementation: agents connect to nearby sensors and store pheromones in there. In the long-term this would be the most powerful solution: sensors, being active, are more reliable and could implement pheromone evaporation autonomously. However, such a solution is presently very costly. Also, wireless sensor networks exhibit battery-exhaustions problems (and thus limited-life) which the RFID solution prevents.

Clearly, the solution that would most closely mimic the actual behavior of social insects would be that of physically releasing markers in the environment. For instance, Svennebring and Koenig [27] have implemented and tested – for the sake of terrain exploration – robots equipped with a pen to leave special metal-ink trails in the pavement, and a sensor to sense such trails. In this way, a group of robots can enforce a simple form of pheromone-based coordination (e.g., if an ink trail is sensed by a robot, it means that another robot has already covered that part of the terrain). Apart from the fact that spreading ink around is not a nice and easily acceptable solution for pervasive computing scenarios, the RFID tag solution is much more flexible, in that it enables using more semantic information for a wider range of applications.

An interesting approach that exploits RFID tags to enforce a sort of marker-based coordination among the activities of robots devoted to collectively construct composite artifacts is described in [31]. On the one hand, robots can acquire awareness about the current position and nature of the building blocks by reading information embedded in RFID tags attached to them. On the other hand, robots that move blocks can write updated location information on them, for other robots in the team to continue the cooperative construction work in a coordinated way. Although focused on a rather different application scenario, such RFID-based form of coordination definitely supports the use of RFID technology as a general substrate for indirect and pheromone-based forms of interactions in the physical world.

9. CONCLUSIONS AND FUTURE WORK

RFID technology, whose effectiveness in improving our interactions with the physical world has already been proved in a variety of pervasive computing projects, also represents a flexible and low-cost way to take advantage of the power and simplicity of pheromone-based interaction models. In particular, the approach presented in this paper exploits RFID tags as a sort of distributed environmental infrastructure that mobile autonomous agents – whether humans or robots – can exploit to spread and sense digital pheromones. Thus, agents can adaptively acquire context-awareness and coordinate with each other in a very simple way, two features that can be fruitfully exploited in a variety of application scenarios.

While a preliminary prototype implementation already shows the feasibility of our approach, a number of research directions are still open to improve its practical applicability. First, more experiments are required to better verify the scalability of the proposed approach to very large-scale scenarios and in the presence of a large number of users. In particular, based on the limitations identified in this paper, effective solutions must be found to the problems related to broken pheromone trails and pheromone evaporation. Second, we need to explore the possibility of extending our strictly local and environment-centered approach (the only exploited information is that available in RFID tags) with the possibility of accessing additional information made available by some sort of "pheromone" servers. The coupling of local RFID pheromone information with some more global information, without undermining the advantages of our approach, can enable reaching higher degrees of context-awareness whenever a network connection is available. Third, we are perfectly aware that the use of RFID to spread and access distributed information in an environment raises serious privacy and security concerns [26], suggesting limiting the use of our system within controlled indoor environments. However, since a number of diverse approaches have been analyzed and proposed to tackle these issues [22], we are confident that suitable solutions will be soon available for integration in our approach.

As a final note, and although this is out of our research competences and goals, we think that a deeper theoretical investigation of the effects of our system in real-world scenarios of use will be needed to better assess its effectiveness and to possibly identify the emergence of phenomena not identified by this paper. As an example, the analyzed effect of pheromone saturation somewhat resembles the effect of information saturation in social and game-theoretic studies [24]. More in general, the study of existing human-to-human indirect interaction models could provide useful insights on the potential impact of our approach [18].

Acknowledgements: work supported by the project CASCADAS (IST-027807) funded by the FET Program of the European Commission.

REFERENCES

[1] G. Avoine, “Bibliography on Security and Privacy in RFID Systems”, MIT Technical Report, 2006.

[2] O. Babaoglu, H. Meling, A. Montresor, “A Framework for the Development of Agent-Based Peer-to-Peer Systems”, International Conference on Distributed Computing Systems, Vienna (A), IEEE CS Press, 2002.

[3] E. Bonabeau, M. Dorigo, G. Theraulaz, “Swarm Intelligence”, Oxford University Press, 1999.

[4] C. Bornhovd, T. Lin, S. Haller, J. Schaper, “Integrating Automatic Data Acquisition with Business Processes: Experiences with SAP’s Auto-ID Infrastructure”, International Conference on Very Large Databases, Toronto (CA), IEEE CS Press, 2004.

[5] G. Collins, “Next Stretch for Plastic Electronics”, Scientific American, 291:74-81, Scientific American Press, 2004.

[6] C. Floerkemeier, M. Lampe, “RFID middleware design - addressing application requirements and RFID”, Smart Objects and Ambient Intelligence Conference, Grenoble (F), 2005.

[7] D. Hähnel, W. Burgard, D. Fox, K. Fishkin, M. Philipose, “Mapping and Localization with RFID Technology”, International Conference on Robotics and Automation, Barcelona (ES), IEEE Press, 2004.

[8] B. Holldobler, E. Wilson, "The Ants", Springer-Verlag, 1990.

[9] K. Greene, “Wireless Wonder Chip”, www.technologyreview.com/printer_friendly_article.aspx?id=17182, 2006

[10] S. Hsi, H. Fait, “RFID Enhances Visitors’ Experience at the Exploratorium”, Communications of the ACM, 48(9):60-65, ACM Press, 2005.

[11] A. Kleiner, J. Prediger and B. Nebel, “RFID Technology-based Exploration and SLAM for Search And Rescue”, International Conference on Intelligent Robots and Systems, Beijing (PRC), IEEE Press, 2006.

[12] V. Kulyukin, C. Gharpure, J. Nicholson, S. Pavithran, “RFID in Robot-Assisted Indoor Navigation for Visually Impaired”, International Conference on Intelligent Robots and Systems, Sendai (J), IEEE Press, 2004.

[13] Q. Li, M. De Rosa, D. Rus, “Distributed algorithms for guiding navigation across a sensor network”, International Conference on Mobile Computing and Networking, San Diego (CA, USA), ACM Press, 2003.

[14] J. McLurkin, J. Smith, “Distributed Algorithms for Dispersion in Indoor Environments using a Swarm of Autonomous Mobile Robots”, International Symposium on Distributed Autonomous Robotic Systems, Toulouse (F), IEEE CS Press, 2004.

[15] M. Mamei, R. Quaglieri, F. Zambonelli, “Making Tuple Spaces Physical with RFID Tags”, Symposium on Applied Computing, Dijon (F), ACM Press, 2006.

[16] M. Mamei, F. Zambonelli, “Spreading Pheromones in Everyday Environments Through RFID Technology”, Symposium on Swarm Intelligence, Pasadena (CA, USA), IEEE CS Press, 2005.

[17] R. Menezes, R. Tolksdorf, “A New Approach to Scalable Linda-systems Based on Swarms”, Symposium on Applied Computing, Orlando, (FL, USA), ACM Press, 2003.

[18] V. Parunak, “A Survey of Environments and Mechanisms for Human-Human Stigmergy”, Environment for Multi-Agent Systems II, LNCS 3830, Springer Verlag, 2006.

[19] V. Parunak, “Go to the Ant: Engineering Principles from Natural Agent Systems”, Annals of Operations Research, Kluwer Academic Publishers, 75:69-101, 1997.

[20] V. Parunak, S. Brueckner, J. Sauter, “Digital Pheromones for Coordination of Unmanned Vehicles”, Environments for Multi-agent Systems I, LNCS 3374, Springer Verlag, 2005.

[21] M. Philipose, K. Fishkin, M. Perkowitz, D. Patterson, D. Fox, H. Kautz, D. Hahnel, “Inferring Activities from Interactions with Objects”, IEEE Pervasive Computing, 3(4):50-57, IEEE CS Press, 2004.

[22] M. R. Rieback, B. Crispo, A. Tanenbaum, “The Evolution of RFID Security”, IEEE Pervasive Computing, 5(1):62-69, IEEE CS Press, 2006.

[23] I. Satoh, “A Location Model for Pervasive Computing Environments”, International Conference on Pervasive Computing and Communications, Kauai Island (HW, USA), IEEE CS Press, 2005.

[24] R. Savit, R. Manuca, and R. Riolo, “Adaptive Competition, Market Efficiency, and Phase Transitions”, Physical Review Letters, 82(10):2203-2206, American Physical Society, 1999.

[25] W. Shen, B. Salemi, P. Will, “Hormone-Inspired Adaptive Communication and Distributed Control for CONRO Self-Reconfigurable Robots”, IEEE Transactions on Robotics and Automation, 18(5):1-12, IEEE Press, 2002.

[26] F. Stajano, “RFID Is X-Ray Vision”, Communications of the ACM, 48(9):31-33, ACM Press, 2005.

[27] J. Svennebring, S. Koenig, “Building Terrain Covering Ant Robots: a Feasibility Study”, Autonomous Robots, 16 (3):313-332, Springer Verlag, 2004.

[28] C. Swedberg, “First Responders Can Tag Victims For Tracking”, RFID Journal, www.rfidjournal.com/article/articleview/2708/1/1, 2006.

[29] R. Want, “Enabling Ubiquitous Sensing with RFID”, IEEE Computer, 37(4):84-86, IEEE CS Press, 2004.

[30] R. Want, “An Introduction to RFID Technology”, IEEE Pervasive Computing, 5(1):25-33, IEEE CS Press, 2006.

[31] J. Werfel, Y. Bar-Yam, D. Rus, R. Nagpal, “Distributed Construction by Mobile Robots with Enhanced Building Blocks”, International Conference on Robotics and Automation, Orlando (FL, USA), IEEE Press, 2006.

[32] D. Weyns, N. Boucke, T. Holvoet, “Gradient Field-Based Task Assignment in an AGV Transportation System”, International Joint Conference on Autonomous Agents and Multi-Agent Systems, Hakodate (J), ACM Press, 2005.