

Pervasive Pheromone-based Interaction with RFID Tags

Marco Mamei, Franco Zambonelli

Abstract—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 application agents is very limited. In this paper, we present a real-world, low-cost and general-purpose, implementation of pheromone-based interaction. This is realized by making use of RFID tags to store digital pheromones, and by having humans and robots to spread/sense pheromones by properly writing/reading RFID tags populating the surrounding environments. We exemplify and evaluate the effectiveness of our approach via an application for object-tracking. This application allows robots and humans to find "forgot-somewhere" objects by following pheromones trails associated with them. In addition, we sketch further potential applications of our approach in pervasive computing scenarios.

Index Terms—Pervasive computing, Pheromone-based coordination, RFID tags

I. INTRODUCTION

Pheromone-based interaction, exploited by social insects to coordinate their activities [BonDT99], has recently inspired a vast number of researches in pervasive and distributed computing systems [BabM02, MenT03, ParBS04, SveK04]. In these works, application agents (e.g., software agents, humans carrying on a PDA, or autonomous robots) interact in an indirect way by leaving and sensing artificial pheromones, 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 [Par97].

Despite the growing interest in pheromone-based interaction, the number of implemented systems exploiting pheromones for coordinating the activities of distributed applications situated in pervasive computing scenarios is very limited. The great majority of the proposals have only been simulated [BabM02, BonDT99, MenT03], only few of them

have been concretely implemented by deploying pheromones in shared virtual data spaces [ParBS04], other few realize pheromones by means of ad-hoc physical markers such as special ink or metal dust [SveK04]. In any case, none of them proposes valid solutions to actually spread pheromones in real-world everyday environments. Discarding centralized – not scalable – solutions, as well as power-hungry and costly sensor networks [EstC02, LiR03], it is not easy to find a suitable – cheap, not intrusive, and at the same time flexible – distributed infrastructure on which to store digital pheromones in pervasive environments.

Inspired by this challenge, we propose a novel approach exploiting RFID technology [Wan04] to enforce pheromone-based interaction in pervasive computing scenarios. The key idea of our approach is to exploit RFID tags dispersed in an environment as a sort of distributed memory in which to store digital pheromones. RFID readers, 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 in-range RFID tags. Clearly, such an approach is extremely low cost and not intrusive, as RFID tags will soon be present in any case, in any environment.

Relying on our simple yet flexible approach, a wide range of application scenarios based on pheromone interaction can be realized, ranging from multi-robot coordination [SveK04] to monitoring of human activities [Phi04]. Here, after having illustrated our approach, we detail and evaluate an application to easily find – by following proper pheromone trails – everyday objects forgot somewhere in our homes.

II. 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 concentration of pheromones. This simple mechanism enables ants to globally self-organize their collective activities in a seemingly intelligent way despite the very limited abilities of individuals of acquiring and processing contextual information in a cognitive way. For this reason, systems of social insects are said to be characterized by “swarm intelligence”, to emphasize the difference with “individual” intelligence [BonDT99, Par97].

Manuscript received November 2, 2005. This work was supported in part by the Italian MIUR and CNR in the “Progetto Strategico IS-MANET, Infrastructures for Mobile ad-hoc Networks”.

Marco Mamei is with the Dipartimento di Scienze e Metodi dell’Ingegneria, Università di Modena e Reggio Emilia, Italy; e-mail: mamei.marco@unimore.it.

Franco Zambonelli is with the Dipartimento di Scienze e Metodi dell’Ingegneria, Università di Modena e Reggio Emilia, Italy; e-mail: franco.zambonelli@unimore.it.

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, they 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 also contributes to re-enforce the pheromone trail in that such ant spreads pheromones in its turn. To some extent, the environment becomes a sort of distributed repository of contextual information holding the paths' information to all the discovered food sources. The natural tendency of the 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 disappears; analogously, when a food source is extinguished, the corresponding pheromone trail disappears because no longer reinforced [BonDT99].

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

- it completely decouples agent (i.e., ant) 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;

- it naturally supports application-specific context awareness, in that pheromones provide a representation of the environment in terms of paths leading to food sources;

- it naturally supports adaptation of activities, in that pheromones represent a contextual information that, when no longer updated, tends to vanish;

- the algorithms underlying pheromone-based interaction are simple and involve only local interactions (each ant locally deposits and follows pheromones without any clue – and associated burden/complexity – 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 [BonDT99], P2P computing [BabM02, MenT03], robotics [ParBS04, SveK04], self-assembly [SheS02], and (as in our approach) pervasive computing, incorporate and exploit pheromone-based interaction mechanisms.

III. DEPLOYING PHEROMONES WITH RFID TECHNOLOGY

At the core of our approach for deploying digital pheromones in an environment is the technology of Radio Frequency Identification (RFID). RFID tags are small wireless radio transceivers that can be attached unobtrusively to objects as small as a watch or a toothbrush. Tags can be purchased off the shelf, cost roughly €0.20 each and can withstand day-to-day use for years, in fact, being battery-free, they 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. 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. The tags respond or store data accordingly using 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 up to some meters.

A. Scenario Assumptions

Our approach requires a scenario in which the operational environment is densely enriched with RFID tags. Tags can be attached at any – even small – object (we refer to these generically as *object-tags*). Also, tags are assumed to be attached at fixed locations (e.g. doors, corridors, etc.) and at unlikely-to-be-moved objects (e.g. beds, washing machines, etc.). We refer to these tags as *location-tags*. Tagging a location or a fixed object involves sticking a RFID tag on it, and making a database entry mapping the tag ID to a name and a spatial location. RFID readers accessing one of these tags can lookup the tag ID into the database and infer to be close to a specific object or location. Thus, beside pheromones, RFID can be used to enforce a simple yet effective localization mechanism for RFID readers (i.e., for the users or robots carrying them) [Hah04, Sat05].

It is worth emphasizing that current trends indicate that (i) within a few years, 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 equipped with a RFID reader). These factors will make our assumption become a de facto situation, and will make our approach become directly deployable at nearly zero cost.

B. Pheromone Deployment

As anticipated in the introduction, pheromones are created by means of data-structures stored in RFID tags. The basic scenario consists of human users and robots carrying handheld computing devices, provided with a RFID reader, and running an agent-based application.

The agent, unobtrusively from the user/robot, continuously detects in range *location-tags* to infer its current location as it roams across the environment. Moreover, the agent controls the RFID reader to write or read on need pheromone data structures (consisting at least in a pheromone ID) in the tags encountered. This process can create digital pheromone trails distributed across the *location-tags*.

More formally, let us call $L(t)$ the set of *location-tags* being sensed at time t . It is easy to see that an agent can infer that the user/robot is moving when $L(t) \neq L(t-1)$. Thus, if instructed to spread pheromone O , the agent will write O in all the $L(t)-L(t-1)$ tags as it moves across the environment.

For the majority of applications a pheromone trail consisting of only the pheromone ID is not very useful. Like

in ant foraging, most applications involve agents to follow 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 go. To overcome this problem, for each pheromone O , agents store in the *location-tags* not only its ID but also an ever increasing hop-counter $C(O)$ associated with O . If an agent decides to spread pheromone O at time t , the agent reads also the counter $C(O)$ in the $L(t)$. If $C(O)$ is not present, the agent sets $C(O)$ to a fixed value zero. Upon a movement, the agent will store O and $C(O)+1$ in the tags belonging to $L(t+1)$ that do not have O or have a lower $C(O)$.

In addition, to support pheromone evaporation (as described later on), each pheromone has also an associated value $T(O)$ representing the time where the pheromone O has been stored.

The above pheromone data structures are stored in the limited memory of RFID tags. RFID tags, other than with an unchangeable unique identifier, are typically provided with an array of cells, each consisting of few bits. We organize such memory by allocating 3 slots for each pheromone. The first slot will hold the pheromone identifier. The second slot will hold the associated counter. The third will hold the above mentioned timestamp.

C. Pheromone Reading and Evaporation

To read pheromones, an agent trivially accesses neighbor RFID *location-tags* reading their memories. Given the result, the agent will decide how to act on the basis of the perceived pheromone configuration.

To realize pheromone evaporation, since the passive nature of RFID tags does not enable them to directly enforce evaporation, we have adopted the following solution. 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 T . If it is so, the agent deletes that pheromone from the tag. This kind of pheromone evaporation leads to two key advantages:

1. Since the data space in RFID tags is severely limited, it would be most useful to store only those pheromone trails that are important for the application at a given time; old, unused pheromones can be removed.
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 leaving the corresponding pheromone trails. This cause old-pheromone trails to be possibly out-of-date, and eventually corrupted.

In this context, it is of course fundamental to design a mechanism to reinforce relevant pheromones not to let them evaporate. With this regard, an agent spreading pheromone O , will overwrite O -pheromones having an older $T(O)$. From these considerations, it should be clear that the threshold T has

to be tuned for each application, to represent the time-frame after which the pheromone is considered useless or possibly corrupted.

IV. PHEROMONE-BASED OBJECT TRACKING

The application we present to exemplify our approach aims at facilitating the finding of everyday objects (glasses, keys, etc.) forgot somewhere in our homes. The application allows everyday objects to leave virtual pheromone trails across our homes to be easily tracked afterwards. Overall, the application works as follows:

1. As from the assumptions, the objects to be tracked are tagged with proper *object-tags*, distinguished from the *location-tags* identifying locations in the environment (object and location tags can be distinguished by their ID).
2. Users (or robots) are provided with a handheld computing device, connected to a RFID reader, and running the object tracking application.
3. The application can detect, via the RFID reader, *object-tags* carried on by the user. Exploiting the mechanism described in the previous section, it can spread a pheromone identifying such objects into the available memory of near *location-tags*.
4. This enables to spread pheromone trails associated with the objects across the *location-tags* of the environment.
5. When looking for an object, a user can instruct the agent to read in-range *location-tags* searching the object's pheromone in their memory. If such pheromone is found, the user can follow it to reach the object current location.
6. Once the object has been reached, if it moves with the user (i.e. the user grabbed it), the application automatically starts spreading again the pheromone associated with the object, to keep consistency with the new object location.
7. This application naturally suits a multi-user scenario where an user (or a robot), looking for an object moved by another user, can suddenly cross the pheromone trail left before by the object.

A. Spreading Object Pheromones

To spread pheromones, the application needs first to understand which objects are currently being carried (i.e. moved around) by the user. To perform this task unobtrusively, it accesses the RFID reader to detect in-range RFID tags once per second.

Let us call $O(t)$ the set of *object-tags* being sensed at time t , $L(t)$ 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 $L(t) \neq L(t-1)$, then the agent can infer that the user picked-up the object O and the object is moving around. 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.

B. 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 crossed a suitable pheromone trail.

There are two alternatives: either $L(t)$ contains only one *location-tag*, or $L(t)$ contains at least two *location-tags* having O -pheromones with different $C(O)$.

In the former 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 case, the agent notifies the user about the fact 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 localization devices, the agent suggests the user to get closer to the location having higher $C(O)$, by naming the location – e.g., walk to the “front door” – and the user has to know how to get there without further help.

In either cases, following the agent advices, the user gets closer and closer to the object by following its pheromone trail, until reaching it.

V. EXPERIMENTS

To assess the validity of our approach and the effectiveness of the object tracking application, we developed a number of experiments, both adopting the real implementation and an ad-hoc simulation (to test on the large scale).

A. Real Implementation Set-Up

The real implementation consisted in tagging places and objects within our department (Figure 1a). Overall, we tagged 100 locations within the building (doors, hallways, corridors, desks, etc.) and 50 objects (books, laptops, cd-cases, etc.). Locations have been tagged with ISO15693 RFID tags, each with a storage capacity of 512 bits (each tag contains 60 slots, 1 byte each, thus it is able to store 20 pheromones). Objects have been tagged with ISO14443B RFID tags, each with a

storage capacity of 176 bits (each tag contains only the object ID).

For users, we exploited HP IPAQ 36xx PDAs, each running Familiar Linux 0.72 and J2ME (CVM – Personal Profile) and provided with a WLAN card and an Inside M21xH RFID reader (Figure 1b).

In addition, mobile robots have been realized by installing one PDA connected to a RFID reader onboard of a Lego Mindstorms robot (www.legomindstorms.com). The IPAQ runs an agent controlling both the RFID reader and the robot microprocessor (Figure 1c).

Finally, a wirelessly accessible server holds a database with the associations between tag IDs and places’ and objects’ description (i.e. ID 001 = Prof. Smith’s office door). The IPAQ can connect, via WLAN, to the database server to resolve the tag ID into the associated description. Each IPAQ runs the described application



Figure 1. (a) Some tagged objects. (b) The test-bed PDA hardware. (c) The Lego Mindstorms robots with a PDA and an RFID reader mounted aboard.

B. Simulation Set-Up

To test more extensively and on the large scale, we realized 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 has been simply simulated by an array of integer values.

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

For the sake of comparison, we 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* algorithm, in which the agents perceive pheromones together with the directions in which they increase. Also, these have been compared with a *blind-search* algorithm, in which agents wander randomly fully disregarding pheromones.

C. Results of the Experiments

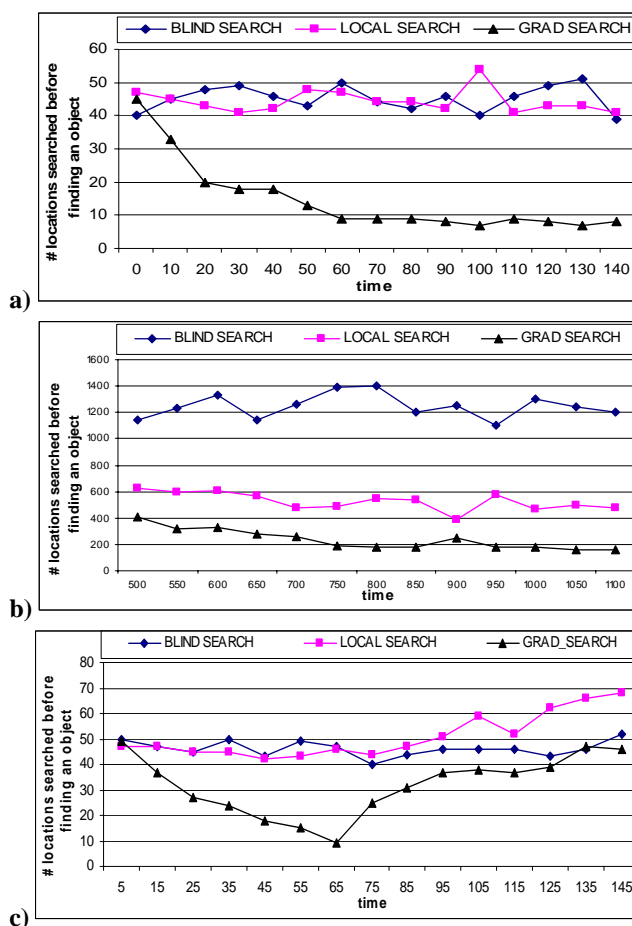


Figure 2. (a-b) Number of places visited before finding a specific object plotted over time in environments consisting of: (a) 100 tagged places, and (b) 2500 tagged places. (c) Number of visited places before finding a specific object plotted over time, when tags tend to saturate.

A first group of experiments 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 consisting in 100 tagged places with 100 objects (Figure 2a); the second consisting in 2500 tagged places with 500 objects (Figure 2b). A number of 10 agents have been simulated to populate these environments wandering around moving objects and spreading pheromones and, at the same time, looking for specific objects. In the experiments, we report the number of places visited before finding specific objects, for different search methods, plotted over time. The reported results are average of about 300 simulations and qualitatively in line with the results obtained – on a smaller scale and on a more limited number of experiments – on the real implementation.

Starting from a scenario free of pheromones (time zero in Figure 2a), the more time passes the more pheromone trails get deployed. *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 less than 10% of the places need to be visited before finding the object. *Local-search*, at least in the small-scale scenario of Figure 2a, 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.

The situation changes when getting to larger-scale scenarios (as in Figure 2b, reporting experiments from a time past the initial transitory). There, both *local-search* and *grad-search* appears reasonably effective. The performance improvement of *local-search* is 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 reader enabling to enforce the *local-search* algorithm only.

A second group of experiments aims at exploring the effects of RFID tag storage saturation upon pheromone spread. This of course represents a big problem, in fact, 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 2c, we report an experiment conducted in the 100-tagged-places-environment described before. This time the tag capacity has been fixed to 50 pheromones (150 bytes), and we plot the number of places visited before finding specific objects, for different search methods, over time. Let us focus 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 already laid down. After some time, *grad-search* works considerably better than *blind-search*, since pheromone trails drive agents. However, as time passes, tags capacity tend to saturate, the objects are moved, but no pheromone trails can be deployed. This situation rapidly trashes performance leading back to *blind-search* performance. In our real implementation (tags with a 512 bits

capacity) the above problem leads to broken trails when more than 30 objects are tracked.

In conclusion, it is rather easy to see that the limited storage capacity of the RFID tags represents a problem for our approach. Basically, if the number of objects to be tracked is greater than the available slots on the RFID tag, in the long run, the problem is unavoidable. Sooner or later, a new object will cross to an already full tag, breaking the pheromone trail. The pheromone evaporation mechanism that we implemented did not help this situation. In another set of experiments, we verified that performance remains more or less the same of Figure 2c, independently on the parameters used to control and tune the evaporation mechanisms.

We still do not have a solution for this problem. Our research 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 would be that the more objects are in the system, the more storage space is available for pheromones, letting the system to scale naturally. The problem is how to manage the fact that *object-tags* containing pheromones can be moved around, breaking the pheromone trail structure. As a partial relief from this problem, it is worth reporting that recent RFID tags have a storage capacity in the order of several of KB, making possible to track hundreds of objects without changing our application.

VI. OTHER APPLICATION SCENARIOS

Pheromone interaction and stigmergy have attracted more and more researches due to their power in supporting agent 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.

In general, RFID tags in an environment and associated to objects can be used to improve *Context-Awareness*. The use of RFID tags as a simple tool for localization (i.e., for location-awareness) has already be outlined. 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, by reading the additional information provided by tags. One of the most interesting work in this direction has been presented in [Phi04]: 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 for writing contextual information; (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 recognize that some tea has been prepared, the teapot start spreading a pheromone trail leading to the fridge and indicating that some tea has already been prepared and is there to cool down.

In line with these ideas is the concept of *Pervasive Workflow Management*. 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 remove the need for a centralized engine in pervasive computing environments and lead to more situated and adaptive scenario. For instance, the RFID tags associated to the items to be processed could store a marker identifying the operations that the item undertook. Workers with RFID readers could simply read the state of the item and process it further. This approach could be employed in traditional manufacturing scenarios as well as in more mundane domestic workflow (e.g. store in the pet's collar a pheromone indicating if it has already eaten or not).

Of course, taking inspiration from the way pheromones are used by ants to let them coordinate their collective movements in an unknown environment toward food sources, we could think at exploiting RFID pheromones to enable a group of users and robots to coordinate on-the-fly their movement in an environment (consider, e.g., a rescue team in a disaster area). For instance, if users spread pheromones around as they walk and are instructed to stay away from existing pheromone trails, one can have reasonable guarantees that the whole environment is explored in an effective way by the group [SveK04].

In this context, it is important to remark that our approach clearly requires the presence of RFID tags before pheromones can be spread. Although RFID tags are likely to be soon densely present in everywhere (embedded in tiles, bricks, furniture, etc.), one cannot rely on this in sensible situations like in a disaster area. In these cases, however, it is possible to conceive solutions where users or robots physically deploy RFID tags on-the-fly while exploring the environment, to be used for subsequent coordination.

VII. CONCLUSIONS

While a preliminary prototype implementation shows the feasibility of our approach, a number of research directions are still open to improve its practical applicability. In particular, more experiments are required to verify the scalability of the proposed architecture to very large-scale scenarios, and more effective solutions must be found to the problem related to broken pheromone trails.

REFERENCES

- [BabM02] O. Babaoglu, H. Meling, A. Montresor, "A Framework for the Development of Agent-Based Peer-to-Peer Systems", Proceedings of the IEEE International Conference on Distributed Computing Systems, Vienna (A), IEEE CS Press, May 2002.
- [BonDT99] E. Bonabeau, M. Dorigo, G. Theraulaz, *Swarm Intelligence*, Oxford University Press (Oxford, UK), 1999.

- [EstC02] D. Estrin, D. Culler, K. Pister, G. Sukjatme, "Connecting the Physical World with Pervasive Networks", *IEEE Pervasive Computing*, 1(1):59-69, Jan.-March 2002.
- [Hah04] D. Hähnel, W. Burgard, D. Fox, K. Fishkin, M. Philipose, "Mapping and Localization with RFID Technology", Proceedings of the IEEE International Conference on Robotics and Automation, Barcelona (ES), IEEE Press, April 2004.
- [LiR03] Q. Li, M. De Rosa, D. Rus, "Distributed algorithms for guiding navigation across a sensor network", Proceedings of the ACM Conference on Mobile Computing and Networking, San Diego, CA (USA), ACM Press, October 2003.
- [MenT03] R. Menezes, R. Tolksdorf, "A New Approach to Scalable Linda-systems Based on Swarms", Proceedings of the ACM Symposium on Applied Computing, Orlando, FL (USA), ACM Press, March 2003.
- [Par97] V. Parunak, "Go to the Ant: Engineering Principles from Natural Agent Systems", *Annals of Operations Research*, 75:69-101, 1997.
- [ParBS04] V. Parunak, S. Brueckner, J. Sauter, "Digital Pheromones for Coordination of Unmanned Vehicles", Proceedings of the 1st International Workshop on Environments for Multi-agent Systems, LNAI 3374, Springer Verlag, 2004.
- [Phi04] M. Philipose, K. Fishkin, M. Perkowitz, D. Patterson, D. Fox, H. Kautz, D. Hähnel, "Inferring Activities from Interactions with Objects", *IEEE Pervasive Computing*, 3(4):50-57, 2004.
- [Sat05] I. Satoh, "A Location Model for Pervasive Computing Environments", Proceedings of 3rd IEEE International Conference on Pervasive Computing and Communications, Kauai Island, HI (USA), IEEE CS Press, March 2005.
- [SheS02] 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, Oct. 2002.
- [SveK04] J. Svennebring, S. Koenig, "Building Terrain Covering Ant Robots: a Feasibility Study", *Autonomous Robots*, 16 (3):313-332, May 2004.
- [Wan04] R. Want, "Enabling Ubiquitous Sensing with RFID", *IEEE Computer*, 37(4):84-86, April, 2004.

Marco Mamei is an associate researcher at the University of Modena and Reggio Emilia, where he received the PhD in computer science. His current research interests include distributed and pervasive computing, swarm intelligence, and self organization. He is a member of the IEEE, AIIA, and TABOO. Contact him at Dipartimento di Scienze e Metodi dell'Ingegneria, Università di Modena e Reggio Emilia, Via Allegri 13, Reggio Emilia, Italy; mamei.marco@unimore.it.

Franco Zambonelli is a professor of computer science at the University of Modena and Reggio Emilia. His research interests include distributed and pervasive computing, multiagent systems, and agent-oriented software engineering. He received his PhD in computer science from the University of Bologna. He is a member of the IEEE, the ACM, AIIA, and TABOO. Contact him at Dipartimento di Scienze e Metodi dell'Ingegneria, Università di Modena e Reggio Emilia, Via Allegri 13, Reggio Emilia, Italy; franco.zambonelli@unimore.it.