A Ubiquitous Knowledge-based System to enable RFID Object Discovery in Smart Environments

M. Ruta, T. Di Noia, E. Di Sciascio, F. Scioscia and E. Tinelli

Politecnico di Bari, via Re David 200, I-70125, Bari, Italy

Abstract. This paper presents an extended framework supported by a suitable dissemination protocol to enable *ubiquitous Knowledge Bases* (u-KBs) in pervasive RFID environments. A u-KB is a distributed and decentralized knowledge base where the factual knowledge (*i.e.*, individuals) is scattered among objects disseminated within the environment, with no centralized repository and coordination.

1 Introduction

In pervasive contexts intelligence is embedded into a physical environment by means of a relatively large number of heterogeneous micro devices such as RFID tags and wireless sensors, each conveying a small amount of useful information. Due to power and cost constraints, they are usually endowed with very low storage space, little or no processing capability and short-range, low-throughput wireless links. Each mobile host in the area can access information only on micro devices in its communication range. Consequently, approaches based on centralized control and information storage are utterly impractical in such scenarios.

In our previous work [1], we devised solutions for the integration of semantic-enhanced EPCglobal RFID into Mobile Ad-hoc NETworks (MANETs). Nevertheless, we were still forced to use a fixed central component for reasoning over a Knowledge Base (KB). This led to expensive information duplication within the environment: semantic annotations were placed simultaneously on tags and within the KB; moreover, the reasoning engine was a single point of failure. Here we want to show how these problems can be mitigated if a more distributed approach is followed. We present a general framework to carry out an advanced matchmaking using metadata stored in RFIDs lacking unique and fixed knowledge bases. An advanced resource discovery is supported by a dissemination protocol allowing to exactly locate suitable descriptions directly on tags attached to objects. The proposed framework comprises the specification of components and operations of a *u-KB* (ubiquitous Knowledge Base), as well as a distributed application-layer protocol for dissemination and discovery of knowledge embedded within RFID tags in a MANET-based computing environment. RFID readers are adopted as cluster-heads w.r.t. tags in their radio range and they are able to automatically build up a multi-hop communication infrastructure when placed in the same area [2]. For lower network layers we adopt IEEE 802.11 [3], IP and UDP, while for knowledge representation we exploit DL-based ontology languages originally conceived for

Ruta M., Di Noia T., Di Sciascio E., Scioscia F. and Tinelli E. (2008).
A Ubiquitous Knowledge-based System to enable RFID Object Discovery in Smart Environments.
In *Proceedings of the 2nd International Workshop on RFID Technology - Concepts, Applications, Challenges*, pages 87-100
DOI: 10.5220/0001741900870100

the Semantic Web effort, particularly DIG [4], which is a more compact equivalent of OWL-DL[1].

The remaining of the paper is organized as follows: in the next Section we present motivation of our paper; in Section 3 some significant theoretical aspects of the proposed approach are revised; in Section 4 we outline the proposed data propagation and retrieval framework, whereas in Section 5 our proposal is explained and motivated by means of a case study. Section 6 closes the paper.

2 Motivation

Motivation for this paper stems from our experience in using Knowledge Representation (KR) approaches –based on Description Logics (DLs)– in pervasive contexts, and particularly in semantic-based discovery frameworks for objects equipped with RFID tags [1]. In traditional applications (see [5, 1.5] for a survey) a Knowledge Representation System (KRS) plays an architectural role very much like a Database Management System (DBMS). Both are used as central repositories where knowledge of the problem domain can be inserted (forming a KB) and upon which automated inference procedures can be executed in order to extract implicit knowledge. Hence, in traditional KRSs, a KB is seen as a single large entity which is immediately available, either in local storage or via a high-throughput network link. This approach is effective only as long as large computing resources and a stable network infrastructure are granted. A different approach is needed to adapt KR tools and technologies to functional and non-functional requirements of mobile and ubiquitous computing applications. They are characterized by user (and device) mobility, dependency on context, severe resource limitations. Hence, knowledge-based systems designed for wired networks are hardly adaptable, due to architectural differences and performance issues.

As Figure 1 shows, in our vision a u-KB layer provides access to knowledge embedded into semantic-enhanced EPCglobal RFID tags populating a smart environment. Discovery and reasoning tasks can then be performed either by hosts in the local MANET or by a remote entity through a gateway exposing a high-level interface (*e.g.*, Web Sevices of RPC (Remote Procedure Call) or REST (REpresentational State Transfer) type) and translating remote methods into operations on the u-KB. This paper focuses on the definition of the components of a u-KB layer, while parallel research effort is being spent into the adaptation of reasoning procedures to resource-constrained mobile devices.

2.1 Application Scenarios

All application areas of RFID technology [6] can be enhanced by a semantically rich description and discovery layer, without depending on a centralized infrastructure. In the lifecycle of industrial products, manufacturing and quality control can exploit accurate descriptions of raw materials, components and processes. Sale depots benefit from

[1] OWL Web Ontology Language, W3C Recommendation, February 10th 2004, available at http://www.w3.org/TR/owlfeatures/

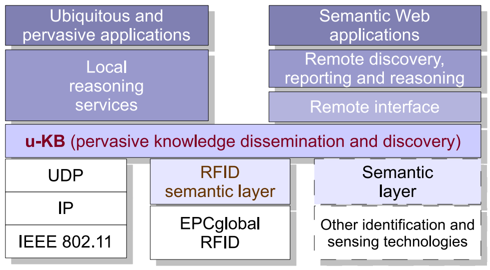

Fig. 1. Architecture of the proposed approach.

easier inventory management and can introduce *ubiquitous commerce* [7] capabilities like in [8], withouth expensive investments in infrastructure. Finally, smart post-sale services can be provided to purchasers, by integrating knowledge discovery in home and office appliances [1].

Asset management is greatly improved in those scenarios where retrieval should depend on object properties and purposes, rather than mere identification codes. In healthcare applications, equipment, drugs and patients can be thoroughly and formally described and tracked, enabling to provide not only monitoring but also decision support to clinicians. Wireless pervasive technologies help break barriers between patient management in the hospital and at home. Likewise, in museums and archaeological sites, smart semantic-based content fruition can be granted to local visitors as well as to remote clients connected through the Internet, leveraging the lightweight infrastructure already deployed for internal inventory.

The u-KB approach can be extended to other monitoring and sensing technologies beyond RFID. Wireless semantic sensor networks [9] are an emerging yet challenging technology. Semantic-based sensory data dissemination and query processing are needed to enable advanced solutions for *e.g.*, precision agriculture and disaster recovery.

3 From KBs to u-KBs

3.1 KB Components and Operations

A DL knowledge base has two components [5]: a **TBox**, containing *intensional* knowledge in the form of an ontology describing general concepts and properties of the reference domain; an **ABox** containing *extensional* knowledge that is specific to the individuals of a particular problem within the domain.

Current KRSs are characterized in terms of what *functions* they provide to applications, instead of exposing system data structures and low-level operations [10]. Two basic functions were identified for KB management. **(1) Tell**: build the TBox and the ABox by explicit assertions of terminological knowledge and information about individuals. **(2) Ask**: extracting (implicit) knowledge by using inference procedures that determine if the meaning of the query is implied by the information that has been told to the system. This paradigm has led to detailed and formal interface specifications for Knowledge Representation Systems (such as *KRSS* [11] and, more recently, *DIG* [4]), implemented by most KRSs. The ability to remove information from a KB is also desirable. Due to technical reasons, most systems allow to *Un-Tell* (*i.e.,* retract) only information that has been previously told explicitly [12]. Nevertheless, experience has shown that, for the vast majority of applications, in production environments the TBox seldom or never changes after an initial knowledge acquisition phase (see [5, ch. 8] for a review).

3.2 u-KB Components

In our approach we preserve the distinction between TBox and ABox. The TBox is contained in an ontology file, which can be managed by one or more mobile hosts. We hypothesize that ontologies are defined before object annotation and u-KB deployment and do not change during normal system activity. We adopt Ontology Universally Unique Identifiers (OUUIDs) [13] to mark ontologies unambiguously and to associate each individual to the ontology w.r.t. it is described.

The ABox is scattered within a smart environment, as KB individuals are physically tied to micro devices deployed in the field. In RFID-based scenarios, each individual is a semantically annotated object/product description, stored within the RFID transponder the object is clung to. Each annotation refers to an ontology providing the intensional knowledge for a particular domain. In detail, each individual is characterized by: globally unique item identifier (the EPC code in the case of RFID tags); OUUID; semantic annotation, stored as a compressed document fragment in the DL-based language DIG; a set of data-oriented attributes, which allow to integrate and extend logic-based reasoning services with application-specific and context-aware information processing. Since several object categories can co-exist within the same physical space, multiple u-KBs can actually populate the same environment and share the system infrastructure.

3.3 u-KB Operations

In our u-KB approach, we adhere to the classic Tell/Ask model. These functions, however, are implemented in a novel way, coping with the characteristics of pervasive computing scenarios. Section 4 defines in detail the data structures and protocol devised to build a u-KB system.

Tell/Un-tell operations are *transparent* to users, *i.e.*, no explicit knowledge declaration/retraction is required. The system allows autonomic creation and maintenance of a "virtual" knowledge base, by means of a data alignment protocol between caches of the various mobile hosts. Each host advertises individuals it sees in its proximity via RFID. Other hosts store advertisements in their cache and forward them to nearby nodes. The protocol keeps track of the freshness of advertised individuals through sequence numbers, so that only updates will be automatically propagated. Each individual has also a limited Time-To-Live (TTL), so that it will be automatically removed (un-told) from the u-KB if not renewed. The system tends toward a steady state where every host is aware of all individuals in the environment.

Ask operations require a preliminary retrieval phase. The requester host specifies the ontology identifier and a range for each particular attribute it is interested in. Access to local cache tables provides IP addresses of hosts owning all the individuals that meet the specified criteria. Pre-filtered “on-demand” provisioning of KB individuals avoids unnecessary data transfers, minimizes transmissions for data alignment and prevents propagation issues in the case of description update or individual removal. The requester can fetch ontology file and filtered individuals from their respective providers, so as to reconstruct a local subset of the whole KB, containing only the TBox and individuals which are actually needed. Then it is able to submit any Ask-type request to a local or remote reasoning engine. In the present work we aim at assuring “on-demand” knowledge availability, so we will not specify further how reasoning is carried out.

4 How to Build a u-KB

The proposed framework presents a two level infrastructure where RFID is exploited at the **field layer** (able to interconnect tags dipped in the environment and readers able to receive the transmitted data) whereas the **discovery layer** is related to the inter-reader ad-hoc communication (see Figure 2). The communication between the tag field and readers exploits the semantic-enhanced EPCglobal RFID protocol data exchange [1], whereas the data propagation among readers is performed following a data dissemination paradigm in 802.11 [3] proposed here.

Thus, the resource discovery is based on three stages: (1) the extraction of good’s parameters (for carrying object characteristics from field layer to discovery one); (2) resource data dissemination (to make the overall nodes fully aware of the “network content”); (3) the extraction of resource annotations (for carrying semantic-based descriptions from field level to the discovery one) for the further matchmaking. Each reader involved in the data propagation and/or in the object discovery, maintains a cache containing the advertisements which will be matched against requests. It plays a central role in the whole service oriented architecture as it advertises contextual parameters

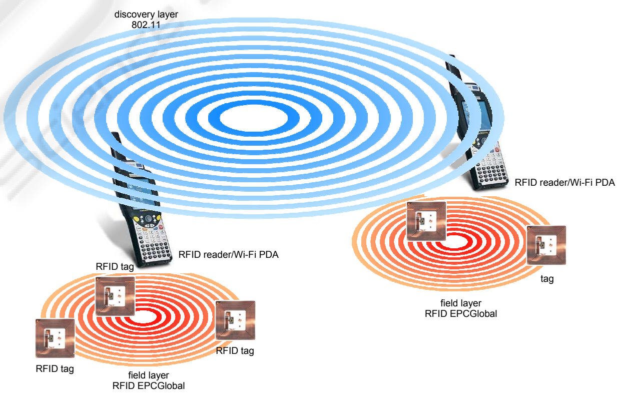

Fig. 2. Field and Discovery layers in the proposed framework architecture.

referred to tags in its radio range (at the field layer) and during the further phase (at the discovery layer), it will receive requests from nearby nodes (via 802.11) and in case it will extract semantic annotations from tags in range (via RFID EPCglobal) so replying to the requester.

We hypothesize each resource in the MANET is labeled by means of the triple [*SOURCE ADDRESS*, *OUUID*, *EPC*], where the first value is the IP address of the RFID reader which has "seen" the resource, the second one marks the specific reference ontology the resource is associated with, the last one is the Electronic Product Code. Initially, each reader will advertise, for each resource, the managed reference OUUID as well as some context-aware parameter (*i.e.*, the resource life time). So the initial selection allows to choose only semantically compatible services (OUUID matching) which have suitable values for context aware parameters.

To perform the data dissemination, resource providers periodically send **advertisement packets** also specifying the maximum number of hops for the advertisement travel (`MAX_ADV_DIAMETER`). The cyclic advertisement diffusion allows to cope with tag and reader mobility. During their travel, the advertisements are forwarded using MAC broadcasts and can be stored in the cache memories of the nodes they go through. When starting a resource matchmaking, a node generally attempts to cover the request by using resource descriptions stored within its own cache memory. If some semantically annotated description is missing, it can be retrieved in unicast using apposite **demand PDUs**. On the contrary, if a requester has no resource descriptions in its cache or if managed resources are considered insufficient to satisfy the request, the node can send a **solicit PDU** with a specified maximum travel diameter (`MAX_REQ_DIAMETER`) in order to get new resource locators. A node receiving a solicit, replies (in unicast) providing cache table entries matching parameters contained within the solicit frame. If it does not manage any information satisfying the solicit, it will reply with a "no matches" message. During their travel, replies to the demand and solicit PDUs are used to update the cache memory of forwarding nodes. Figure 3 and Figure 4 show the typical sequence and involved actors of the data dissemination phase and of semantic annotations retrieval, respectively.

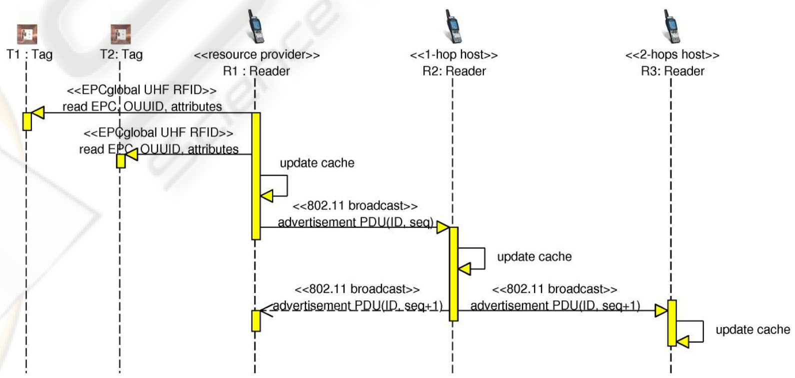

Fig. 3. Tag data dissemination phase.

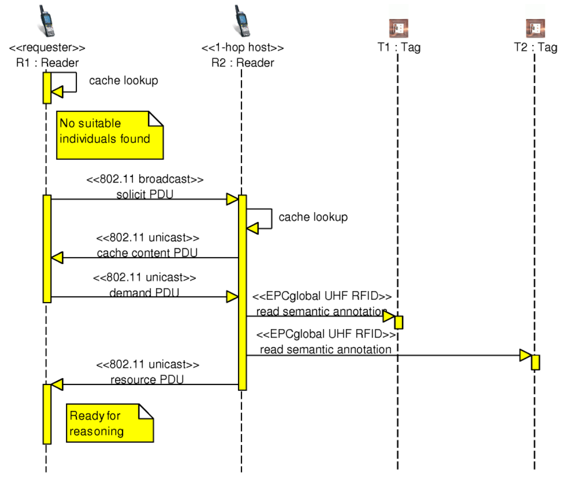

Fig. 4. Retrieval of advertised semantic annotations from tags.

Table 1. PDU types exploited in the proposed framework.

TYPE	BIT SET	PDU
A	0	Advertisement
B	1	Cache entry
C	2	Solicit
D	3	Demand
E..L	4..7	reserved

4.1 Advertisement PDU

In Figure 5 the structure of an advertisement PDU is sketched. All the resources inventoried by a reader are advertised by means of a unique advertisement PDU and then the size of the packet increases proportionally with the number of tags in the reader range. In what follows we outline PDU fields.

- `TYPE`: the kind of PDU (see Table 1).
- `FLAGS`: it contains one status flag to distinguish the kind of transmission (uni or broadcast); the remaining flags are reserved for future purposes.
- `TRAVELED HOPS`: the number of hops already traversed by the packet. A reader sets this value to 1 and it is increased every time a node forwards the packet.
- `NUMBER OF RESOURCES`: how many tags are in the reader radio range.
- `ADVERTISEMENT ID`: the reader's sequence number.
- `NODE SEQUENCE NUMBER`: the sequence number of the node forwarding the packet. If the packet has been sent by a reader this field value coincides with the previous one.
- `SOURCE ADDRESS`: the IP address of the reader.
- `RESOURCE PARAMETERS`: a composite, variable length field depending on the number of advertised resources. In particular, it contains the OUUID value, the remaining life time of a resource, the maximum hops number for the advertisement travel, and finally the EPC code of the single tag.

Table 2. Constant values used in the framework.

NAME	MEANING	VALUE
DEFAULT_RTIME	Time interval between two consecutive advertisement packet transmissions	30000 ms
POLLING_TIME	Time a reader node waits for the echo of the advertisements	7500 ms
MAX_ADV_JITTER	Maximum value for random time waited when forwarding advertisement packets	600 ms
ONE_HOP_WAIT	Timer set by a requester node after sending a solicit packet, waiting for cache contents reception	2000ms
HOP_TRAVERSAL_TIME	Time a node needs to process and forward a solicit packet sent by a neighbor	50ms
ACK_RTT	Timer set by a requester node waiting for ack after a solicit has been sent	50 ms
DISCOVERY_DIAMETER	Current search diameter (in hops) during discovery phase	4
MAX_RETRIES	Maximum number of retransmissions before a reader assumes there are no neighbors	5

A reader which has inventoried a tag series, broadcasts an advertisement every DEFAULT_RTIME milliseconds (see Table 2 for constant values). Nearby nodes forward the packet by broadcasting it to their neighbors; as a consequence, the reader listens to the echo of the advertisement packet it originally transmitted. Thus, it can obtain a confirmation of the presence of other nodes in its neighborhood. If the reader does not receive any echo within POLLING_TIME milliseconds (less than DEFAULT_RTIME), it will retransmit the advertisement, assuming that a collision or a transmission error has occurred. After MAX_RETRIES retries it can be assumed there are no neighbors, so the transmission of the advertisement can be scheduled after a longer timeout in order to reduce power consumption.

When a node receives an advertisement, it extracts information about the resources and, in case of "new" items, it adds cache entries; otherwise, before updating stored data, the node verifies if the received information is more recent or has ran across a shorter path than the existing one. This happens also when multiple readers attempt to scan the same tag. If the cache is updated and the maximum advertisement diameter has not been reached, then the advertisement is forwarded; otherwise the whole packet is silently discarded. This simple mechanism grants each mobile node in the network sends the same advertisement at most once. Furthermore, in order to reduce the collision probability (recall that MAC 802.11 protocol does not provide any acknowledgment frame for broadcast transmission), each host waits a random time t ($t \in [0,$ MAX_ADV_JITTER$]$) before transmitting.

We developed an analytical model of the data dissemination protocol (not reported here due to lack of space) to estimate whether the inclusion of the 96-bit EPC code in each resource advertisement could lead to an unacceptable network overhead. Let us consider a typical scenario with a partition-free network of 30 readers and 1000 tags per reader. Since typical read rates are not above 100 tags/s for EPCglobal Gen-2 UHF RFID [14], we deem that reasonably the advertisement period DEFAULT_R_TIME should not be set below 30 s; moreover, let us set MAX_ADV_DIAMETER=4. In that case, total traffic generated by data dissemination has an upper bound of ≈ 590 kB/s, *i.e.*, $\approx$ 20 kB/s per reader. Furthermore, generated traffic is a linear function of tag population size, thus granting a theoretically acceptable scalability to the system.

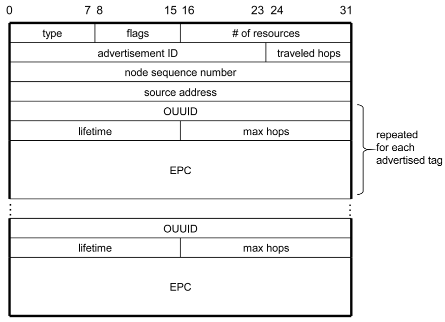

Fig. 5. Advertisement PDU.

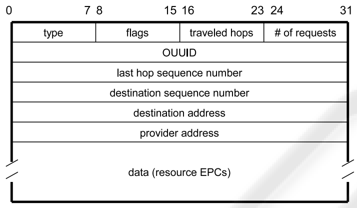

Fig. 6. Demand PDU.

4.2 Demand PDU

When starting a matchmaking, a node must firstly look within its cache table for entries compatible with the request and, in case, it must require in unicast the corresponding semantically annotated descriptions from their respective owners. A demand PDUs as the one sketched in Figure 6 is used. In what follows the meaning of introduced PDU fields is summarized.

- `TYPE`: it is set to 3.
- `FLAGS`: analogous to the corresponding field of the advertisement PDU.
- `TRAVELED HOPS`: how many hops the frame has already gone across.
- `LAST HOP SEQUENCE NUMBER`: the sequence number of the last node processing the request.
- `DESTINATION SEQUENCE NUMBER`: the sequence number of the destination node.
- `DESTINATION ADDRESS`: the address of the last node processing the request.
- `PROVIDER ADDRESS`: is the address of the destination node.
- `OUUID`: the ontology's unique identifier.
- `NUMBER OF REQUESTS`: the number of requested resource descriptions.
- `DATA`: the size of this field depends on the number of requests; it contains the EPCs of the tags whose descriptions are required.

4.3 Solicit PDU

Basically, the soliciting mechanism is analogous to the advertising one. Figure 8 shows the format of a solicit packet. Not reported PDU fields are analogous to the previous ones.

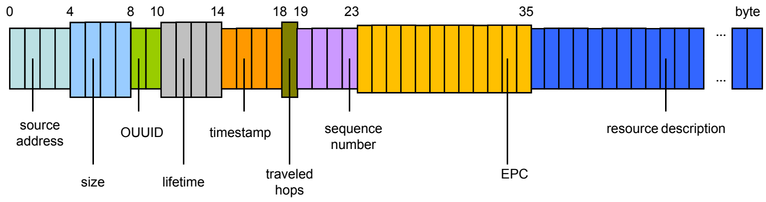

Fig. 7. The structure of a record in the reader cache.

- `TYPE`: it is set to 2.
- `FLAGS`: it maintains the ordinary structure and functionality.
- `TRAVELED HOPS`: hops the packet has already gone across.
- `TOTAL HOPS`: total hops the PDU has to skip. Together with the *TRAVELED HOPS* field, it regulates the frame travel.
- `REQUEST ID`: unambiguously labels the PDU in order to distinguish different solicit requests.

A node generating a solicit packet waits for an acknowledgment from each neighbor for `ACK_RTT` seconds. By means of this frame the requester elicits information about nearby nodes: it can exactly know the number of neighbors. Each node located at `DISCOVERY_DIAMETER` hops from the requester, after receiving a solicit PDU, replies with a cache content PDU in unicast toward the node the solicit cames from. Readers receiving a cache content PDU update their own cache and recursively sends back a cache content PDU, till the original requester node receives information it needs.

4.4 Cache Table Management

Each reader manages a cache table where it stores characteristics of both tags in its radio range and tags it has "seen" in the network. Figure 7 shows the structure of a typical entry. Here the content of each field is reported.

- `Source address`: the address of the resource provider.
- `Size`: the description size (in byte).
- `OUUID`: a numeric identifier for the specific ontology.
- `Lifetime`: the remaining time to live of a resource/tag.
- `Timestamp`: it marks the last reference to the entry (read/write).
- `Traveled hops`: distance (hops number) between provider and cache holder.
- `Sequence number`: it is referred to the last resource provider.
- `EPC`: the Electronic Product Code of a specific resource/tag.
- `Resource description`: the semantically annotated description of a resource. It will have a variable length, but in some case there could be a pointer to the compressed file containing the DIG description.

An entry could be added to the cache table whenever the node receives an advertisement or a cache content frame arrives. Note that if a node receives an advertisement frame, the corresponding entry of the cache table is updated either if the sequence number within the packet is higher than the one stored or if the route the PDU suggests is shorter than the previously stored one.

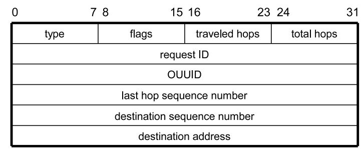

Fig. 8. Solicit PDU.

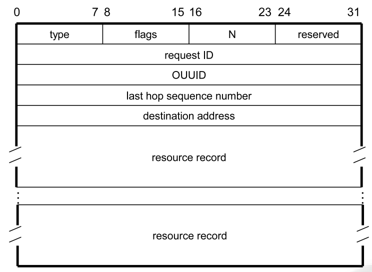

Fig. 9. Cache content PDU.

A cache content packet (whose format is depicted in Figure 9) has a variable length according to the number of resource handles the PDU transports. PDU fields are outlined in what follows.

– `TYPE`: it is set to 1.
– `FLAGS`: it maintains the ordinary meaning.
– `N`: the number of resources handles (and then cache tuples) the packet transports.
– `REQUEST ID`: is the identifier of the original request.
– `OUUID`: the identifier of the reference ontology.
– `LAST HOP SEQUENCE NUMBER`: the sequence number of the node sending the packet.
– `DESTINATION ADDRESS`: requester IP address.

Last fields are the resource records.

5 Case Study

5.1 RFID-based u-KB

The proposed framework has been studied specifically in pervasive computing environments where a wide range of objects/products are endowed with RFID transponders conforming to the EPCglobal standard for class I - second generation UHF tags [15]. Mobile RFID readers equipped with IEEE 802.11 wireless connectivity (hereafter *hosts*) are responsible for u-KB creation and management. Tagged objects represent KB individuals in our system. In our previous work [1] the EPCglobal standard was enhanced to support storage and retrieval on RFID tags of an ontology identifier, a set of attributes and a compressed semantic-based annotation, in addition to the EPC identifier. These basic elements recur in our definition of an individual in a u-KB, as explained above. For the sake of conciseness, here we omit details of the semantic-enhanced RFID protocol proposed in [1] so assuming the reader be familiar with it.

Table 3. SELECT command able to detect only semantic enabled tags.

PARAMETER	Target	Action	MemBank	Pointer	Length	Mask
VALUE	100_2	000_2	01_2	00010101_2	00000010_2	11_2
DESCRIPTION	SL flag	set in case of match, clear otherwise	EPC memory bank	initial address	number of bits to compare	bit mask

Table 4. READ command able to extract OUUID from the TID memory bank.

PARAMETER	MemBank	WordPtr	WordCount
VALUE	10_2	000000010_2	00000010_2
DESCRIPTION	TID memory bank	starting address	read up to 2 words (32 bits)

5.2 Interactions with EPCglobal RFID Technology

1. Dissemination. After each advertisement period `DEFAULT_TIME`, a host scans RFID tags in its range. Only semantic enabled tags are preselected, by means of a *Select* command with parameters as shown in Table 3.

EPC codes of semantic based tags are then scanned individually and TTL of corresponding cache table entries are refreshed. If the host detects a new EPC code not present in its cache table, it will read its OUUID and contextual attributes, exploiting two *Read* commands, as shown in Table 4 and Table 5 respectively. Data extracted from the RFID tag will be stored in a new cache entry with a fresh sequence number. Conversely, if the EPC code is not detected for an existing local entry in the cache table, the host will wait for the TTL to expire before removing the entry. This prevents the well-known issue of "RFID event flickering"[2] [16] from causing incorrect removal/addition of u-KB individuals. At the end of the loop, the cache table is fully updated and the host can issue an *advertisement* PDU to notify individuals to neighboring hosts.
2. Discovery. When a host receives a *demand* PDU for which it is the destination, it starts an RFID scan of semantic enabled tags only, as seen above. During inventory, for each detected EPC among those listed in the PDU payload, it reads the semantically annotated compressed object description stored in the User memory bank of the tag, with a *Read* command as in Table 6. Finally, the host replies to the requester.
3. Ontology Provisioning. As already suggested in [1], the EPCglobal Object Naming Service (ONS)[3] can be used as a fallback mechanism for ontology support in ubiquitous computing contexts if an Internet connection is available.
4. System Evaluation. After a feasibility study using IBM WebSphere RFID middleware, we are currently performing an extensive simulation campaign in *ns-2*[4] of the full protocol stack, in scenarios with both fixed and moving readers. We expect valuable insight for a thorough evaluation of the proposed approach, in particular w.r.t. network load, duration of service discovery sessions, hit ratio (percentage of successful resource retrieval) and sensitivity to topology variations (due to RFID reader and tag mobility). Due to lack of space, we are not able to report here our early findings.

---

[2] Due to collisions, a tag might not be detected in every consecutive scan. This phenomenon can trigger spurious leave-enter event pairs.
[3] Object Naming Service (ONS - v. 1.0), EPCglobal Ratified Specification, Oct. 4, 2005, http://www.epcglobalinc.org
[4] ns-2, the network simulator, http://www.isi.edu/nsnam

Table 5. READ command to extract contextual attributes from User memory bank.

PARAMETER	MemBank	WordPtr	WordCount
VALUE	11_2	000000000_2	00001000_2
DESCRIPTION	user memory bank	starting address	read up to 8 words (16 bytes)

Table 6. READ command to extract the compressed semantic annotation from User memory bank.

PARAMETER	MemBank	WordPtr	WordCount
VALUE	11_2	000001000_2	00000000_2
DESCRIPTION	user memory bank	starting address	read up to the end

6 Conclusions

Building on our previous work, we have presented an approach to carry out an advanced matchmaking using semantic metadata stored in RFID tags without unique and fixed knowledge bases. An advanced resource discovery framework is supported by a knowledge dissemination protocol, so allowing an "on-demand" retrieval of suitable descriptions directly from tags located on the objects. An analytical model has been developed for a preliminary assessment of resource requirements, while functionality tests have been carried out in a reduced simulation environment. We are currently working on a thorough evaluation of the approach using *ns-2* simulation environment.

Acknowledgements

We wish to acknowledge support of Apulia project PE_074 "IC Technologies for tracking of agricultural and food products equipped with RFID tags".

References

1. Ruta, M., Di Noia, T., Di Sciascio, E., Scioscia, F., Piscitelli, G.: If objects could talk: A novel resource discovery approach for pervasive environments. International Journal of Internet and Protocol Technology (IJIPT), Special issue on RFID: Technologies, Applications, and Trends 2 (2007) 199–217
2. Ramanathan, R., Redi, J.: A brief overview of ad hoc networks: Challenges and directions. IEEE Communications Magazine 40 (2002) 20–22
3. 802.11, I.: Information Technology Telecommunications and Information Exchange between Systems Local and Metropolitan Area Networks Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. ANSI/IEEE Std. 802.11, ISO/IEC 8802-11. First edn. (1999)
4. Bechhofer, S., Möller, R., Crowther, P.: The DIG Description Logic Interface. In: Proceedings of the 16th International Workshop on Description Logics (DL'03). Volume 81 of CEUR Workshop Proceedings. (2003)
5. Baader, F., Calvanese, D., Mc Guinness, D., Nardi, D., Patel-Schneider, P.: The Description Logic Handbook. Cambridge University Press (2002)
6. Weinstein, R.: Rfid: A technical overview and its application to the enterprise. IT Professional 07 (2005) 27–33

7. Watson, R., Pitt, L., Berthon, P., Zinkhan, G.: U-Commerce: Expanding the Universe of Marketing. Journal of the Academy of Marketing Science 30 (2002) 333–347
8. Di Noia, T., Di Sciascio, E., Donini, F., Ruta, M., Scioscia, F., Tinelli, E.: Semantic-based bluetooth-rfid interaction for advanced resource discovery in pervasive contexts. International Journal on Semantic Web and Information Systems (IJSWIS) 4 (2008) 50–74
9. Ni, L., Zhu, Y., Ma, J., Li, M., Luo, Q., Liu, Y., Cheung, S., Yang, Q. In: Semantic Sensor Net: An Extensible Framework. Springer Berlin / Heidelberg (2005) 1144–1153
10. Levesque, H.: Foundations of a Functional Approach to Knowledge Representation. Artificial Intelligence 23 (1984) 155–212
11. Patel-Schneider, P., Swartout, B.: Description-Logic Knowledge Representation System Specification. (KRSS Group of the ARPA Knowledge Sharing Effort)
12. Brachman, R., McGuinness, D., Patel-Schneider, P., Resnick, L., Borgida, A.: Living with CLASSIC: When and how to use a KL-ONE-like language. Principles of Semantic Networks (1991) 401–456
13. Ruta, M., Di Noia, T., Di Sciascio, E., Donini, F.: Semantic-Enhanced Bluetooth Discovery Protocol for M-Commerce Applications. International Journal of Web and Grid Services 2 (2006) 424–452
14. Kawakita, Y., Mistugi, J.: Anti-collision performance of Gen2 air protocol in random error communication link. In: Proceedings of the International Symposium on Applications and the Internet Workshops - SAINT 2006. (2006) 68–71
15. Traub, K., Allgair, G., Barthel, H., Bustein, L., Garrett, J., Hogan, B., Rodrigues, B., Sarma, S., Schmidt, J., Schramek, C., Stewart, R., Suen, K.: EPCglobal Architecture Framework. Technical report, EPCglobal (2005)
16. Römer, K., Schoch, T., Mattern, F., Dübendorfer, T.: Smart Identification Frameworks for Ubiquitous Computing Applications. Wireless Networks 10 (2004) 689–700