Semantic-Based RFID Data Management

Roberto De Virgilio, Eugenio Di Sciascio, Michele Ruta, Floriano Scioscia, and Riccardo Torlone

Abstract Traditional Radio-Frequency IDentification (RFID) applications have been focused on replacing bar codes in supply chain management. Leveraging a ubiquitous computing architecture, the chapter presents a framework allowing both quick decentralized on-line item discovery and centralized off-line massive business logic analysis, according to needs and requirements of supply chain actors. A semantic-based environment, where tagged objects become resources exposing to an RFID reader not a trivial identification code but a semantic annotation, enables tagged objects to describe themselves *on the fly* without depending on a centralized infrastructure. On the other hand, facing on data management issues, a proposal is formulated for an effective *off-line* multidimensional analysis of huge amounts of RFID data generated and stored along the supply chain.

1 Introduction

A supply chain is a complex system composed by organizations and people with their activities involved in transferring a product or service from a supplier to a final customer. The key to make a successful supply chain relies on an extended collaboration, implying the integration among actors involved in the productive and logistic network. An integrated and flexible management of logistics (physical and information flows) has to be set-up both inside and outside factory boundaries. Specialized production and distribution processes suffer from the limited interactions allowed by rigid networks. As a result, nowadays a relevant component of competition in the market occurs among logistical chains. The supply chain can no longer be represented as static or linear, but it needs to be evaluated dynamically,

R. De Virgilio (✉)
Universitá Roma Tre, Rome, Italy
e-mail: dvr@dia.uniroma3.it

D.C. Ranasinghe et al. (eds.), *Unique Radio Innovation for the 21st Century*,
DOI 10.1007/978-3-642-03462-6_6,

as a complex system made of interactions and connections among actors operating along the chain.

Many empirical investigations have demonstrated there is a positive correlation between enterprise performances and its propensity and attitude to be integrated into larger systems. This is the reason why enterprises are more and more attentive to the opportunity offered by both coordination and cooperation among their internal functions and the other external actors contributing in different ways to the business. Hence, information has been an increasingly strategic asset in the last few years. It covers a determinant position in both logistics and marketing. The physical flow of raw materials, products and their related information is considered as a strategic element for quality standards of products/services, for business analysis and evaluation and finally to allow corrective actions. In particular, current trends in the consumer products market assign a growing significance to retailers in the governance of supply chains. The growing dimension of retail groups – sustained by the reached degree of concentration in that field – increases their power against producers and makes them privileged centers of value accumulation, acting as filters for the information flow for the whole chain. As a result, the main retailers are investing in new technology in order to boost the information exchange and are mandating the adoption of interoperable solutions to commercial partners (Smith 2005; Karkkainen 2003).

Radio-Frequency IDentification (RFID) is an AutoID (automatic identification) technology, interconnecting via radio two main components: (1) transponders (commonly named *tags*) carrying data, located on the objects to be identified; (2) interrogators (also known as *readers*) able to receive the transmitted data. Benefits introduced by RFID technology w.r.t. barcodes include: (i) unlike optical scan, alignment between reader and tag is not needed; (ii) longer read range (up to few meters); (iii) nearly simultaneous detection of multiple RFID tags; (iv) higher tag storage capacity (up to several kBs Ayoub Khan and Manoj 2009). Because of these features, RFID provides better levels of automation in the supply chain and helps prevent human errors (e.g., a reader can inventory an entire shipment in one pass while it is loaded into a warehouse, without having to scan each product manually). In latest years, industry is progressively rallying around few worldwide standards for RFID technologies. In this effort a leading role is played by the EPCglobal consortium EPC global. However, two main issues restrain a more advanced exploitation of RFID capabilities. Firstly, the original identification mechanism only enables a trivial string matching of ID codes, providing exclusively "yes/no" replies. Furthermore, RFID-based technology usually relies on a stable support infrastructure and fixed information servers where massive data analysis is quite difficult without the support of proper data management and aggregation schemes. Recall that an accurate evaluation of enhancements in RFID-based supply chains rely on global trend inspections over the supply chain itself which requires multidimensional analyses of huge amounts of RFID data generated and stored in central DBMSs. Serious data management issues are then inevitably inherited and they must be faced on.

In this chapter an innovative model for supply chain management is presented, aiming to overcome these limitations by adopting the Ubiquitous Computing (ubicomp) paradigm. As originally introduced by Mark Weiser (1999), ubicomp requires both information and computational capabilities to be deeply integrated into common objects and/or actions and the user will interact with many computational devices simultaneously, exploiting data automatically extracted from "smart objects" permeating the environment during his/her ordinary activities.

Leveraging a distributed architecture, the model provides a unified framework for both quick run-time analysis (with respect to a local fragment of the overall infrastructure) and stand-alone massive business logic elaborations (with respect to a centralized DBMS) following needs and requirements of the supply chain actors. An extension of current RFID technology supporting logic-based formalisms for knowledge representation is exploited. Semantic-based object/product annotations are stored into RFID tags, exploiting machine-understandable ontological languages originally created for the Semantic Web effort and based on Description Logics (DLs) (Bargide), such as RDF (Resource Description Framework)[1] and OWL (Web Ontology Language)[2] Semantically rich and unambiguous information is allowed to follow a product in each step of its life cycle. The model allows to trace and discover the information flow –associated to products thanks to their RFID tags– along the supply chain, and to formalize various supply chain analyses. Different perspectives can be so followed (e.g., product-centric, node-centric, path-oriented, time-oriented). Exploiting semantic-based queries, product and process information can be read, updated and integrated during manufacturing, packaging and supply chain management, thus allowing full traceability up to sales, as well as intelligent and de-localized interrogation of product data.

The chapter is structured as follows: Sect. 2 highlights the benefits of semantic-based approach for RFID data management in a typical scenario and Sect. 3 surveys relevant related work. Section 4 reports on basics of formalism and notation exploited in the detailed framework presentation of Sect. 5. Finally, Sect. 6 recalls some experimental evaluation corroborating the approach and Sect. 7 concludes the chapter.

2 Motivating Scenario

A simple reference example should clarify our approach, also highlighting its benefits. Let us suppose to monitor the life cycle of an apparel item, a *cotton shirt*, and to follow every production step surveying and extracting relevant product/process

---

[1] RDF Primer, W3C Recommendation, February 10th 2004, available at http://www.w3.org/TR/rdf-primer/.

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

information. Each production stage will see the progressive joining of annotations to enhanced RFID tags attached – for instance – to cotton yarn containers shipped to the factory (first production stage), shirt pallets (logistic step) and single product packages (final sale phase). Because of traceability requirements, a tag will store: (1) quantitative data pertaining to the product besides the Electronic Product Code (EPC) identifier; (2) high-level qualitative information about production or delivery/logistics processes, expressed as semantic annotations w.r.t. a reference ontology of the specific industry domain. Information extracted via RFID can be used for a variety of purposes. First of all – at each stage of the product evolution – accurate verifications can be performed about expected quality requirements of the product/process. Moreover, intelligent deliveries can be routed from warehouse to different production departments according to their specific characteristics.

A product can inherit (relevant parts of) the semantically annotated description of its raw materials, through properties defined in the reference ontology for the relevant domain. Further product attributes can also be stored on the RFID tag, such as size, production date and (for perishable products) expiration date. Finally, location, entering and exiting times are stamped by each supply chain actor conveying the item, such that advanced applications can be enabled. Beyond rather basic features allowed by a traditional data-oriented usage of RFID, a semantic-based approach makes possible further interactions.

A relevant aspect of the approach is that the semantic-enhanced RFID technology allows to share information, so optimizing the supply chain and improving performance both in terms of logistics features and by providing innovative services available for all involved actors. The envisioned framework can support a range of use cases, involving different stakeholders along a product life cycle. Several tangible (economic) and intangible benefits are expected. During product manufacturing and distribution, a wide-area support network interconnecting commercial partners is not strictly needed. This is a significant innovation with respect to common RFID supply chain management solutions (De et al. 2004).

Semantic-enabled RFID tags contain a structured and detailed description of product features, endowed with unambiguous and machine-understandable semantics. Goods auto-expose their description to any RFID-enabled computing environment is reached. This favors decentralized approaches in order to offer context-aware application solutions, based on less expensive and more manageable mobile ad hoc networks. In addition to improved traceability, a semantic-based approach provides unique value-added capabilities. By combining standard and non-standard inference services devised in (Di Noia et al. 2004), several semantic-based matchmaking schemes can be designed to meet goals and requirements of specific applications. Adopting a logic-based approach, query flexibility and expressiveness are much greater than both keyword-based information retrieval and standard resource discovery protocols, which support code-based exact matches only. This enables an effective query refinement process and can increase user trust in the discovery facility. Semantic-enhanced RFID object discovery can be leveraged also for sales and post-sale services, by assisting customers in using their purchased products more effectively.

3 Related Work

In the latest few years automatic identification technologies are gaining more and more interest. This is mostly due to their possible use in many industrial applications. AutoID systems allow the exchange of information about moving people, animals or goods for tracking in real time. Now RFID is the fastest growing sector of the AutoID business (Raza 1999).

RFID benefits in supply chains have been widely acknowledged in both distribution and warehousing sectors. In latest years, they are becoming evident also in the retail and post-sales domains. RFID-derived benefits include timeliness, accuracy and completeness (Karkkainen 2003). Wal-Mart, the world leading retailer, has been at the forefront of RFID experimentation, leveraging its retail power to mandate RFID adoption amongst selected suppliers (Pepe and Risso 2008; Tajima 2007). Significant retailer cost savings associated with RFID-monitored short-shelf-life products are achieved and the introduction of RFID technology by retailers might improve their supply chain efficiency, accuracy and security (Jones et al. 2004; Angeles 2005). De et al. (2004) introduced a reference system for real time tracking of items in a ubiquitous context. That work constitutes a research prototype of current technological architectures for supply chain management, endorsed by worldwide special interest groups such as the EPCglobal consortium (Traub et al 2005). A further relevant example outlines a retail shopping scenario with RFID technology allowing the consumer to engage in a seamless shopping experience (Sellitto 2007).

The benefits of RFID have always been associated with the powerful and dynamic automated data acquiring capabilities of the technology (Smith 2005). Nevertheless, RFID has received relatively little investigation from the informative standpoint. RFID technology should actually be viewed as an information facilitator that can directly improve the decision-making capabilities of personnel within an organization and information sharing across boundaries of partner organizations. Currently, researchers appear to overlook the important process of using RFID-derived information, which is a significant factor in deriving benefits such as visibility of the supply chain, product traceability and retailer's inventory monitoring. In other words, supply chains should be considered not only as product flow networks, but also as information flow networks. Effective exploitation of the supply chain information infrastructure can increase business process awareness and thus improve both performance analysis and support to decision processes.

Innovative information technology solutions are required for RFID-based data storage and analysis in order to enable such a new kind of smart supply chain. With reference to business processes, for deep analysis with respect to great lapse and huge amounts of data, the most interesting research field is in defining new storage models for a more expressive and sophisticated description of RFID data. There are two main directions.

The first one concerns run-time processing of data streams (Bai et al. 2006; Bai et al. 2007 Wang et al. 2006; Jeffery et al. 2006). Most current approaches, however, track only very basic information, namely *raw data* produced by RFID readers.

Raw data consist in *(EPC, location, time)* triples, where EPC is a unique product identifier, while location and time mark each RFID reading event. More advanced information and knowledge representation techniques have not been significantly integrated yet within RFID technology in smarter supply chain management solutions able to support analyses with higher-level semantics.

The second research direction focuses on off-line computation and efficient data storage (Wang and Liu 2005; Ban et al. 2005; Gonzalez 2006) Wang et al. (2005) formalized some features and semantics of RFID events, proposing an extension of the Entity-Relationship model. Based on such extended conceptual model, data streams are analyzed w.r.t. temporal aspects. Bai et al. (2007) studied limits of using SQL to detect temporal events in a database and illustrated a SQL-like language to query such events in an efficient way. Ban et al. (2005) presented a location-oriented indexing which traces paths registered by RFID readers through a novel representation model. The exploited indexing schema is named *Time Parameterized Interval R-Tree*, as a variant of common R-Tree. Gonzalez et al. (2006) provided instead a new storage model borrowed from the datawarehouse literature. Its core idea is to group items moving together so that the multidimensional analysis can be based on dimensions and measurements like in a typical datawarehouse.

For quick chain interactions aiming at an object/group discovery given a semantically annotated request, we refer to the Semantic Web initiative and adapt techniques and technologies to RFID-based supply chains. The basic goal is to fully characterize products equipped with RFID tags by means of semantic languages such as RDF, OWL or DIG. Semantic Web technologies allow a formalization of annotations in a machine understandable way, so promoting interoperability. A range of tools can be used for information processing and analysis, including rule-based systems, logic-based inference engines and query engines based on declarative languages like SPARQL[3]. The validity of the Open World Assumption (OWA) enables meaningful analyses even in the presence of incomplete information. This feature allows to overcome shortcomings of widely adopted "closed world" paradigms –such as the relational model– that often arise when interfacing heterogeneous information systems of independent partner organizations. This is indeed the case of supply chain management architectures. By means of formal ontologies, knowledge about a specific domain can be modeled and exploited in order to derive new implied information from the one stated within metadata associated to each resource Di Noia (2004).

Few proposals for semantic-based annotation of physical products can be found in literature. A solution applied to ubiquitous commerce environments was introduced in Maass and Filter (2006). RFID tags, however, stored only a product code, which was used as a key to retrieve the corresponding RDF annotation from a central backend information system. This "virtual counterpart" approach Pömel et al. (2006), inherited from traditional RFID applications in supply chain management,

[3]SPARQL Query Language for RDF, W3C Recommendation 15 January 2008, http://www.w3.org/TR/rdf-sparql-query/

poses major architectural and organizational challenges for information sharing in complex multi-party supply chains. Conversely, our core idea is that, as physical products flow among supply chain partners, ipso facto relevant high-level information about them is conveyed and can be exploited for meaningful business analysis at different levels.

Finally the graph-based nature of semantic models produces relevant challenges. Although simple and general, these models cannot be used in their basic form as storage models. Many proposals can be found in literature concerning alternative logic organizations to efficiently analyze semantic data (Marcus et al. 2007). The contribution, however, is based on optimizations strongly dependent on the physical structures used (i.e., dependent on tools that support these structures). Our proposal is enriched by the provisioning of data models that describe semantic information at different levels of abstraction (not only at the physical one). Therefore the optimization is independent from any physical environment.

4 Preliminaries

In order to make the chapter self-contained, hereafter some details about adopted formalisms and languages will be provided.

4.1 Semantic-Based Matchmaking

Semantic-based object discovery is grounded on Description Logics (DLs) which provides the Ontology Web Language (OWL-DL) semantics. Furthermore, there is a strict correspondence between the OWL-DL syntax and the DIG (Description Logic Implementation Group) one which is exploited as interface for HTTP-based reasoners (Bechhofer et al. 2003). Particularly, when facing on implementation issues, DIG 2.0 formalism should be adopted to express both requests and resource descriptions, because it is less verbose and more compact, a mandatory requirement in mobile ad hoc applications. Anyway, here we formalize examples by adopting DL syntax for the sake of readability.

DLs are a family of logic formalisms for Knowledge Representation (Baader et al. 2002) whose basic syntax elements are *concept names*, *role names*, *individuals*. Intuitively, concepts stand for sets of objects in the domain, and roles link objects in different concepts. Individuals are used for special named elements belonging to concepts. Formally, concepts are interpreted as subsets of a domain of interpretation δ, and roles as binary relations (subsets of $\Delta \times \Delta$).

DL formulas give a semantics by defining the interpretation function $\cdot^{\mathcal{I}}$ over each construct. For example, if A and D are two generic concepts, their conjunction $A \sqcap D$ is interpreted as set intersection: $(A \sqcap D)^{\mathcal{I}} = A^{\mathcal{I}} \cap D^{\mathcal{I}}$, and also the other boolean connectives $\sqcup$ and $\neg$, when present, give the usual set-theoretic interpretation of union and complement.

Concepts can be used in *inclusion assertions* $O \sqsubseteq D$, and *definitions* $O \equiv D$, which impose restrictions on possible interpretations according to the knowledge elicited for a specific domain. A DL theory (a.k.a. *TBox* or *ontology*) is basically a set of inclusion assertions and definitions. A *model* of a TBox $\mathcal{T}$ is an interpretation satisfying all inclusions and definitions in $\mathcal{T}$. Many other constructs can be defined, so increasing the expressiveness of the DL. Nevertheless, this usually leads to a growth in computational complexity of inference services (Brachman and Levesque 1984). Hence a trade-off is necessary.

The core idea of the Semantic Web initiative (Berners-Lee et al. 2001) is to annotate information by means of markup languages, based on XML, such as RDF, RDFS and OWL. These languages have been conceived to allow machine understandable, unambiguous representation of Web contents through the creation of domain ontologies, increasing openness and interoperability in the WWW. The strong relationship between DLs and the above referenced languages (Baader et al. 2003) is also evident in the classification of the OWL sub-languages.

- *OWL-Lite*. It allows class hierarchy and simple constraints on relations between classes.
- *OWL-DL*. Based on DLs theoretical studies, it allows a great expressiveness keeping computational completeness and decidability.
- *OWL-Full*. It has a huge syntactic flexibility and expressiveness. This freedom is paid in terms of no computational guarantee.

In this chapter we will refer to the *Attributive Language with unqualified Number restrictions and Concrete Domains* ($\mathcal{ALN}(D)$) DL, a subset of OWL-DL, which has a polynomial complexity both for standard and non-standard inferences. Constructs of $\mathcal{ALN}(D)$ DL are reported in what follows:

- $\top$, *universal concept*. All the objects in the domain.
- $\bot$, *bottom concept*. The empty set.
- A, *atomic concepts*. All the objects belonging to the set A.
- $\neg A$, *atomic negation*. All the objects not belonging to the set A.
- $C \sqcap D$, *intersection*. The objects belonging to both C and D sets.
- $\forall R.C$, *universal restriction*. All the objects participating in the R relation whose range are all the objects belonging to C set.
- $\exists R$, *unqualified existential restriction*. At least one object participating in the relation R.
- $(\geq n\ R)$[4], $(\leq n\ R)$, $(= n\ R)$[5], *unqualified number restrictions*. Respectively the minimum, the maximum and the exact number of objects participating in the relation R.

[4]Notice that $\exists R$ is equivalent to $(\geq 1R)$

[5]We write $(= n\ R)$ for $(\geq n\ R \sqcap \leq n\ R)$

- f, *concrete features*. An extension to basic DLs that allows to link concepts to a concrete domain D (e.g., integers, reals, time and so on) through a set of unary predicates p. Each concrete feature f can be expressed as $p(f)$ with $p : \delta \Rightarrow D$, where δ is the feature domain. In this chapter only the concrete domain of integers and the following unary predicates will be considered: $(\geq_k\ g), (\leq_k\ g), (=_k\ g)$, with g a feature and k an integer value.

Knowledge Representation (KR) approaches to matchmaking usually exploit classical deductive services, namely *classification* (also known as *subsumption*) and *consistency* (i.e., *satisfiability*). Basically, given a request/resource pair annotated w.r.t. a common reference ontology, classification allows to check whether all request specifications are included within the resource description. Whereas consistency verifies whether some specifications in the request contradicts (some of) the ones within the resource annotation. In both cases, the response is then a binary *true/false* value. Although these inference services are very useful in the early phases of a discovery process, they are not sufficient to rank a set of resources with respect to a request.

Given R (for Request) and O (for Offer) both consistent with respect to an ontology $\mathcal{T}$, logic-based approaches to matchmaking proposed in literature (Paolucci et al. 2002; Li and Horrocks 2004) use classification and consistency to grade match results in five categories:

- *Exact*. All the features requested in R are exactly provided by O and vice versa –in formulae $\mathcal{T} \models R \Leftrightarrow O$.
- *Full-Subsumption*. All the features requested in R are contained in O –in formulae $\mathcal{T} \models O \Rightarrow R$.
- *Plug-In*. All the features offered in O are contained in R –in formulae $\mathcal{T} \models R \Rightarrow O$.
- *Potential-Intersection*. There is an intersection between the features offered in O and the ones requested in R –in formulae $\mathcal{T} \not\models \neg(R \wedge O)$.
- *Partial-Disjoint*. Some features requested in R are conflicting with some offered in O –in formulae $\mathcal{T} \models \neg(R \wedge O)$.

While exact and full matches seldom occur, a user may get several potential and partial matches. Then a logic-based matchmaker should provide a – logic – ranking of available resources w.r.t. the request, but what we get using classification and consistency is a boolean answer. Also partial matches might be just "near miss" (e.g., maybe just one requirement is in conflict), but a pure consistency check returns a hopeless *false* result, whereas it could be interesting to retrieve "not so bad" resources according to their similarity to the request.

Let us consider concepts R and O and an ontology $\mathcal{T}$. If a partial match occurs, i.e., if they are not compatible with each other with respect to $\mathcal{T}$, one may want to retract some specifications in R, G (for *Give up*), to obtain a concept K (for *Keep*) such that $K \sqcap O$ is satisfiable in $\mathcal{T}$.

In Colucci et al. (2003) the Concept Contraction problem was first defined as:

– *Concept Contraction*. Let $\mathcal{L}$ be a DL, R, O be two concepts in $\mathcal{L}$ and $\mathcal{T}$ be a set of axioms in $\mathcal{L}$, where both R and O are satisfiable in $\mathcal{T}$. A *Concept Contraction Problem* (CCP), identified by $\langle \mathcal{L}, \mathcal{R}, \mathcal{O}, \mathcal{I} \rangle$, consists of finding a pair $\langle G, K \rangle \in \mathcal{L} \times \mathcal{L}$ such that $\mathcal{T} \models R \equiv G \sqcap K$, and $K \sqcap O$ is satisfiable in $\mathcal{T}$. Then K is a *contraction* of R according to O and $\mathcal{T}$.

  If nothing can be kept in R during the contraction process, we get the worst solution –from a matchmaking standpoint– $\langle G, K \rangle = \langle R, \top \rangle$, that is give up everything of R. Conversely, if $R \sqcap O$ is satisfiable in $\mathcal{T}$, that is a potential match occurs, nothing has to be given up and the solution is $\langle \top, R \rangle$. Hence, a Concept Contraction problem amounts to an extension of a satisfiability one. Since usually one wants to give up as few things as possible, some minimality criteria in the contraction must be defined Gärdenfors (1988).

  If the offered resource O is a potential match for R, it is necessary to assess what should be hypothesized H in O in order to completely satisfy R and then move to a full match. In Di Noia et al. (2003) the Concept Abduction problem was first defined as:

– *Concept Abduction*. Let $\mathcal{L}$ be a DL, R, O be two concepts in $\mathcal{L}$, and $\mathcal{T}$ be a set of axioms in $\mathcal{L}$, where both O and R are satisfiable w.r.t. $\mathcal{T}$. A *Concept Abduction Problem* (CAP), identified by $\langle \mathcal{L}, R, O, \mathcal{I} \rangle$, is to find a concept $H \in L$ such that $\mathcal{T} \models O \sqcap H \sqsubseteq R$, and moreover $O \sqcap H$ is satisfiable in $\mathcal{T}$. We define H a *hypothesis* about O according to R and $\mathcal{T}$.

If $O \sqsubseteq R$ then we have $H = \top$ as a solution to the related CAP. Hence, Concept Abduction amounts to extending subsumption. On the other hand, if $O \equiv \top$ then $H \sqsubseteq R$.

4.2 RFID Data Representation

Whatever RFID application usually generates a tuple stream in the form of a triple (E, l, t), where:

- E is an *EPC* (Electronic Product Code), i.e., a unique identifier stored in a tag and associated to each tagged object;
- l is the *location* where an RFID reader has scanned an object having the E EPC;
- t is the *time* when the reading took place.

As a single tag may have multiple readings at the same location – thus producing a great amount of raw data – cleaning techniques have to be applied. The most used compression converts raw data in *stay records* in the form: (E, l, t_{in}, t_{out}) where t_{in} is the time when the object enters the location l, and t_{out} is the time when the object leaves it. Although this basic solution reduces the amount of data to be stored (even if not considerably), previous data representation loses object transitions information. So an alternative representation of RFID data has been proposed in Lee and

Chung (2008). It involves *trace records* and has the form:

$$E : l_1[t_{in}^1; t_{out}^1] \Rightarrow \ldots \Rightarrow l_k[t_{in}^k; t_{out}^k]$$

where:

- $l_1, \ldots, l_k$ are locations along the path followed by the tag with E EPC ;
- t_{in}^i is the entering time at location l_i;
- t_{out}^i is the exiting time from the location l_i;
- the sequence is ordered by t_{in}^i.

The drawback of such data representations is that they are path-dependent, and therefore only path queries over objects moving together can benefit from them (e.g., *find the average time for jackets to go from tailor's to stores in Rome*). To overcome this limitation, the notion of *entry records* is introduced, which describes product information that can be used in multidimensional analysis. Entry records have the form:

$$E : [A_1, v_1], [A_2, v_2], \ldots, [A_n, v_n]$$

where:

- A_i describes an attribute representing the object with E EPC;
- v_i is the value associated to A_i for this object.

Notice that: (i) an entry record can be used to represent collections of RFID data at different level of details (e.g. raw data or stay records) and (ii) aggregates are based on different combinations of attributes A_i.

4.3 Supply Chain Indexing

In supply chain management, a basic need is to analyze object transitions. A product with an RFID tag can cross many locations in a chain. Tracing its movements, transitions can be expressed as a path $l_1, \ldots, l_n$ in a graph describing the supply chain itself. Different approaches and techniques have been proposed for supply chain indexing in order to effectively compute the path of a tag. Currently they are supported by DBMS physical optimizations. In Ban et al. (2005) and Gonzalez et al. (2006), EPC data features are exploited to group tags and arrange them through bitmap indexes. Lee and Chung (2008) devised an alternative encoding scheme that assigns to each path a pair (*Element List Encoding Number (ELEN) - Order Encoding Number (OEN)*). By assigning a prime number to each node in the chain, ELEN is obtained by multiplying path nodes (with related values) among them. In this way, the path followed by a tag is computed as factorization of the integer assigned to the path itself. Primality of assigned numbers guarantees the correctness of the result. Additionally, OEN is a value able to encode the arrangement among nodes in the path.

Both the above solutions present some non-negligible drawback. Approaches in Ban et al. (2005) and Gonzalez et al. (2006) are dependent on DBMS optimizations and on the similarity assumption between EPCs, whereas framework in Lee and chung (2008) does not cope with computational complexity of prime factorization of an integer. Since a supply chain can present several levels (typically from four to ten), to the best of our knowledge, no efficient algorithms are available for very large values.

To efficiently manage the object transitions, an elementary method was introduced in De Virgilio et al. (2009) for encoding each path $l_1, \ldots, l_n$. The basic idea is to model a supply chain s by a directed *sc-graph* G_s whose nodes are locations of s and there is an edge from a node l_1 to a node l_2 if there is some movement of objects from l_1 to l_2 in s. The *source nodes* of a sc-graph, i.e., the nodes having no incoming edge, usually represent the place where objects are produced, whereas *target nodes* of a sc-graph, i.e., nodes having no outgoing edge, are usually the final stores where products are sold. Then, a *token* is associated to each possible path from a source to a target node. Finally, the encoding is performed in each node n of a sc-graph by assigning to n the set of tokens representing each path from a source to a target node traversing n. The supply chain in Fig. 1 shows as an example the proposed path encoding.

4.3.1 Data Compression

In order to aggregate queries over a large amount of RFID data, it is useful to compress them with respect to different *aggregation factors*. For instance, considering location, data can be aggregated either at city, region or country levels. Similarly, a product can be grouped by brand, category or price.

Let us consider a set of attributes $S = \{A_1, A_2, \ldots, A_n\}$ describing available entry records. By borrowing a typical OLAP (on-line Analytical Processing) approach, attributes are grouped in S according to different *factors* such as location, time or product. Moreover, each factor allows to build a set $R = \{(x, r, y) : x, y \in S\}$,

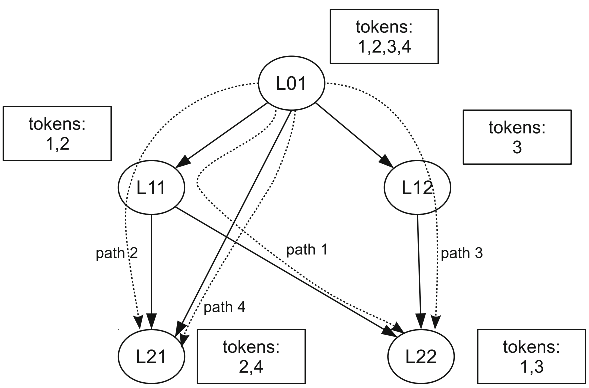


Fig. 1 Supply chain indexing

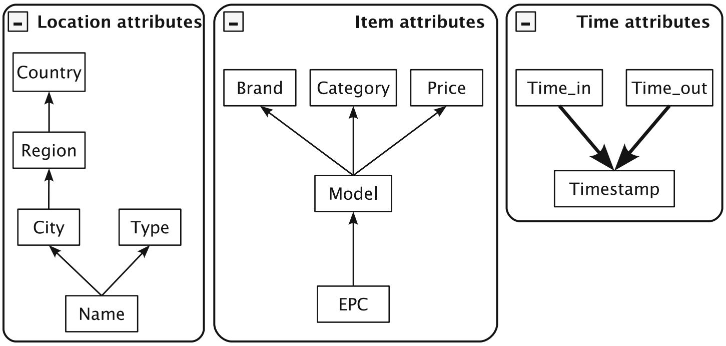


Fig. 2 A taxonomy example

where r is a binary relationship between attributes such as *is-a*, *part-of*, and so on. In this way *taxonomies* can be built starting from the original attribute set. Figure 2 shows possible taxonomies: thin arrows represent *part-of* relationships, thick ones represent *is-a* ones between attributes.

The set R, augmented with taxonomies defined over proper factors in R, suggests interesting attribute aggregations. An *aggregation factor* is defined as a propositional logic expression φ involving attributes occurring in $A_1, \ldots, A_n$. It can be easily noticed that φ depends on taxonomies defined over them. For instance, if $A_i \wedge A_j$ occurs in φ and (A_i, part-of, A_j) then the conjunction can be replaced by A_j, because A_j has a higher aggregation degree higher than A_i. Hence, a set of rules can be defined to simplify an aggregation factor using taxonomies. With respect to the set of attributes in Fig. 2, possible aggregation factors for that domain are: $\varphi_1 = (location \wedge time\_in \wedge time\_out)$ and $\varphi_2 = (model)$. Intuitively, φ_1 aggregates records for items entering and leaving the same location at the same time, while φ_2 aggregates records for items which are similar among them.

Given two entry records $r_1 = \{E_1; [A'_1, v'_1], \ldots, [A'_n, v'_n]\}$ and $r_2 = \{E_2; [A''_1, v''_1], \ldots, [A''_m, v''_m]\}$, an *aggregated record* r_A has the form: $\{E_{r_A}; [r_1, r_2]\}$. Two entry records r_1 and r_2 are aggregated w.r.t. an aggregation factor φ if r_1 and r_2 *satisfy* φ as follows.

Definition 4.1 (φ-satisfiability) *Given an expression φ over a set of attributes $A_1, \ldots, A_n$, and two entry records $r_1 = \{E_1; [A_1, v'_1], \ldots, [A_j, v'_j], \ldots, [A_m, v'_m]\}$ and $r_2 = \{E_2; [A_1, v''_1], \ldots, [A_j, v''_j], \ldots, [A_m, v''_m]\}$, r_1 and r_2 satisfy φ if, by replacing each $\mathrm{A_i}$ in φ with the atom $(v'_i = v''_i)$ for $1 \leq i \leq n$, the obtained formula is true.*

This notion can be used to introduce the one of aggregation based on a given aggregation factor.

Definition 4.2 (φ-aggregation) *Given an aggregated record r_A, r_A is a φ-aggregate if for each pair of entry records r_1, and r_2 occurring in r_A:r_1 and r_2 satisfy φ.*

Hence, an aggregation $\boldsymbol{F}$ can be obtained by a sequence of aggregation factors $\varphi_1, \varphi_2, \ldots, \varphi_n$. Basically, in a supply chain an aggregated record is an item *stock*.

Object movement can be seen as a collection of stocks that split moving from sources to targets. Under this assumption, a useful definition is reported hereafter.

Definition 4.3 (Subsumption) *Given two aggregated records* $r_{A_1} = \{E_{r_{A_1}}; [r_1, r_2, \ldots, r_n]\}$ *and* $r_{A_2} = \{E_{r_{A_2}}; [r'_1, r'_2, \ldots, r'_m]\}$*,* r_{A_1} *is subsumed by* r_{A_2}*, denoted by* $r_{A_1} \lhd r_{A_2}$*, if each* r_i *occurring in* r_{A_1} *also occurs in* r_{A_2}*.*

An object transition can be then seen as a movement from a stock to another one. Hence, an *observation* is defined as an object transition event and it can be represented by a pair (s_1, s_2), where s_1 is a stock including a set of items coming from the stock s_2. It ensues that $s_1 \lhd s_2$.

5 Framework and Approach

Analyses related to a supply chain could be basically intended as involving two different aspects: *on-product* and *batch* control. The former is performed on-line and aims at an object (product) discovery, given features explicitly stated in a request by either controller or customer. The latter happens off-line and refers to a higher level abstraction related to product batches flow. Both kinds of analysis help various actors involved in a supply chain (from producers to retailers to customers) in taking under control large amounts of goods with their related properties as well as to easily track and retrieve specific things or groups of things.

This section describes an integrated EPCglobal RFID-based framework where tagged goods can be either individually and automatically discovered in a pervasive computing scenario or aggregated for enabling centralized massive analysis involving off-line data storage. Both components will be examined in detail below.

5.1 Online Object Discovery

On-line object discovery requires an in-depth comparison between product characteristics and user requirements. In order to quickly produce useful results, advanced (possibly automated) matchmaking techniques are required where both user's request and resource characterization are expressed in a machine understandable format, which allows the needed expressiveness while keeping an acceptable computational complexity. Due to space, power and cost constraints, RFIDs are still currently endowed with low storage, no processing capability and short-range, low-throughput wireless links. Furthermore, each mobile reader in the field can access information only on micro devices in its range. As a consequence, approaches based on centralized control and unique information storage are impractical. On the other hand, when effective network infrastructures are lacking and exploited devices are resource-constrained, the discovery process can be strongly enhanced by exploiting KR techniques and technologies. Semantic-based resource annotation and matchmaking, as well as logic-based ranking and explanation services (Di Noia, et al. 2004), seriously improve resource retrieval experience. Furthermore, the enhancements to EPCglobal RFID standard protocol let users interact with the system without requiring dependable wired infrastructures while hiding technicalities from them.

In the research paper Di Noia (2008) we introduced a knowledge-based variant of EPCglobal RFID, whose primary goal was to keep a backward compatibility with the original technology as much as possible. Protocols to read/write tags were preserved, maintaining original code-based access, in order to ensure compatibility and smooth coexistence of new semantic-based object discovery applications and legacy identification and tracking ones. In our framework we refer to RFID transponders conforming to the EPCglobal standard for class I - second generation UHF tags (Traub 2005) (we assume the reader be familiar with basics of this technology). The proposal was tested in a simulation campaign and in application case studies, showing the benefits of the approach for stakeholders involved each stage of the product lifecycle (Ruta et al. 2007) – from raw materials to production, retail and post-sale services. Here an evolution of the approach is presented.

The practical feasibility of advanced semantic-based usage of RFID technologies must take into account some important constraints. First of all, the severe bandwidth and memory limitations of current RFID systems, in order to meet cost requirements for large-scale adoption. Due to technological advances and growing demand, passive RFID tags with greater memory amounts are expected to be available (Ayoub Khan and Manoj 2009). Nevertheless, XML-based ontological languages like OWL and DIG are far too verbose for a direct storage on RFID tags. Moreover, a mechanism is clearly required to distinguish semantic enabled tags from standard ones, so that semantic based applications can exploit the new features without interfering with legacy applications.

To enable the outlined enhancements, RFID tags and the air interface protocol must provide read/write capabilities for semantically annotated product descriptions with respect to a reference ontology, along with additional data-oriented attributes. Neither new commands nor modification to existing ones have been introduced. To accomplish that, we extend the memory organization of tags compliant with the above referenced standard.

Contents of TID memory area up to $\mathtt{1F}_\mathrm{h}$ bit are invariable. For tags having class identifier value $\mathtt{E2}_\mathrm{h}$ stored in the first byte of the TID bank, optional information could be stored in additional TID memory from $\mathtt{20}_\mathrm{h}$ address. There we store:

- a 16 bit word for optional protocol features, stored starting from $\mathtt{20}_\mathrm{h}$ address most significant bit first: currently only the most significant bit is used to indicate whether the tag is semantic-enabled or not; other bits are reserved for future uses;
- a 32-bit *Ontology Universally Unique Identifier* (OUUID) marking the ontology with respect to the description stored in the tag is expressed Ruta et al. (2006).

In this way, a reader can easily distinguish semantic based tags by means of a *Select* command with parameter values as in Table 1. Values for the triple $\langle MemBank, Pointer, Length\rangle$ identify the bit at $\mathtt{20}_\mathrm{h}$ address in the TID memory bank. The reader commands each tag in range to compare it with bit mask 1_2. The match

Table 1 *Select* command able to detect only semantic enabled tags

Parameter	Target	Action	MemBank	Pointer	Length	Mask
Value	100_2	000_2	10_2	00100000_2	00000001_2	1_2
Description	SL flag	Set (if match)	TID bank	Initial address	Bit to be compared	Bit mask

outcome will be positive for semantic enabled tags only. The *Target* and *Action* parameter values mean that in case of positive match the tag must set its *SL* flag and clear it otherwise. The following inventory step will skip tags having SL flag cleared, thus allowing a reader to identify only semantic enabled tags. Protocol commands belonging to the inventory step have not been described, because they are used in the standard fashion.

In order to retrieve the OUUID stored within a tag, a reader will exploit a *Read* command by adopting parameter values as in Table 2. *MemBank* parameter identifies the TID memory bank and the *WordPtr* value specifies that the reading must start from the third 16-bit memory word, i.e., from 20_h address. Finally, the *WordCount* parameter indicates that 32 bits (two 16-bit words) have to be read.

Contextual parameters (whose meaning may depend on the specific application) are stored within the *User memory bank* of the tag. There, we also store the product annotation. To overcome storage space limitations, it is encoded with a specialized compression algorithm designed for XML-based ontological languages. An RFID reader can perform extraction and storing of a description from/on a tag by means of one or more *Read* or *Write* commands, respectively. Both commands are obviously compliant with the RFID air interface protocol. Table 3 reports parameter values of the *Read* command for extracting the full contents of the User memory, comprising both contextual parameters and the compressed annotation.

The EPCglobal standard also provides a support infrastructure for RFID applications by means of the so called *Object Naming Service* (ONS) EPCglobal (2005). In our approach the ONS mechanism is considered as a supplementary system able to grant the *ontology support*. If a reader does not manage the ontology which provides terminology for the description within the tag, we may retrieve it exploiting the ONS service. The *EPCglobal Network Protocol Parameter Registry* maintains all the registered service suffixes (*ws* for a Web service, *epcis* for a EPCglobal Information Service (providing authoritative information about objects associated with an EPC code), *html* for a Web Page of the manufacturer). We hypothesize to register the novel *dig* suffix to indicate a provisioning service for ontologies with a specified OUUID value.

Table 2 *Read* command able to extract the OUUID from the TID memory bank

Parameter	MemBank	WordPtr	WordCount
Value	10_2	000000010_2	00000010_2
Description	TID memory bank	Starting address	Read 2 words (32 bits)

Table 3 READ command able to extract the semantically annotated description from the User memory bank

Parameter	MemBank	WordPtr	WordCount
Value	11_2	000000000_2	00000000_2
Description	User memory bank	Starting address	Read up to the end

5.2 Off-Line Batch Analysis

An off-line supply chain data analysis basically involves the following steps.

1. *Import* a reference supply chain. It takes as input the topology of a given supply chain in terms of a sc-graph $G(V,E)$ where nodes V correspond to locations and edges E to movements between locations. The graph is imported into a relational DBMS and indexed.
2. An expert user sets an attribute taxonomy and defines the aggregation factors according to queries suggested by stakeholders.
3. Raw data are collected from RFID readers in the field. In a preprocessing phase, raw data are compressed into stay records and ordered by t_{in}. Then aggregation factors defined by the user are incrementally applied to generate a set of aggregated entry records, which are materialized and indexed. In this way a compressed data set is imported into the RDBMS.
4. Finally, users can submit queries to a front-end interface and inspect the answers provided by the system.

In the following subsections we present in greater detail the aggregation process and illustrate the proposed storage schema for aggregated records able to guarantee an efficient query processing.

5.2.1 Data Aggregation

Given an aggregation $F = [\varphi_1, \varphi_2, \ldots, \varphi_n]$ and a stay records sequence *SR* ordered by t_{in}, F is applied to *SR* to generate a set of aggregated entry records *ARS*. Notice that an aggregation makes sense if it is locally applied. In other words, only records having the same location are aggregated. Since information details are contained within tagged items, there is no need for a centralized backend for the whole supply chain. Each supply chain partner has visibility of the relevant attributes and entry records of items passing through its premises and can use its own support infrastructure for off-line data analysis. For each node, aggregation can be then performed according to the most relevant factors for business analysis, selected according to specific management goals. This approach also solves the problem of change of item "ownership" when items flow among different partners (and even across country borders) or they are redistributed in different packages.

Table 4 Entry records aggregation

Record	EPC	Model	Price	Count	Loc	t_{in}	t_{out}
r_1	clt01	Polo	160	1	L12	4	6
r_2	clt02	Polo	160	1	L12	4	6
r_3	clt03	Suit	400	1	L12	4	6

Hereafter a straightforward aggregation strategy is outlined.

1. Records characterized by the same location are grouped in a set L.
2. The aggregation factors φ_i are sequentially applied to elements in L: if there exists an aggregated record rd_A such that current element R' of L and rd_A satisfy υ, then R' can be inserted into rd_A; otherwise a new aggregated record from R' is created.

For example, let us consider three entry records representing clothes (*i.e.*, items) moving into the supply chain shown in Figure 1, as in Table 4.

Now, let us consider an aggregation $F = [\varphi_1, \varphi_2]$ resulting from factors $\varphi_1 = (time\_in \wedge time\_out)$ and $\varphi_2 = (model)$. With respect to φ_2 two aggregated records are produced: $r_{A_1} = \{r_1, r_2\}$ and $r_{A_2} = \{r_3\}$. Finally, applying φ_1, the aggregated record $r_{A_3} = \{r_{A_1}, r_{A_2}\}$ is obtained.

The final step involves building observations, assigning a token to each of them. The following algorithm illustrates that process. It takes as input: a sequence of aggregated records *ARS* increasingly ordered by *time_in*, the chain C in terms of a sc-graph $C(V_C, E_C)$ and a map *LM* where a list of tokens is assigned to each location. The output is a map *OM* whose observations have a given token.

1. Starting from the last record *AR* of *ARS*, go back to find the aggregated record that subsumes *AR*.
2. If it does not exist, a new observation is generated, the list of tokens corresponding to *AR* location is extracted from *LM* and the first token is assigned to the new observation.
3. Otherwise, a new observation is generated where $AR \lhd AR'$. The token to be assigned results from the intersection between the lists of tokens of both locations in AR and AR' excluding the ones coming from other paths reaching AR' location.

5.2.2 Data Storage

Figure 3 shows the relational schema to contain RFID data processed as described above. `PATHMAP` and `LOCATION` store chain information. Each entry in `PATHMAP` has an external reference `LOC_ID` to `LOCATION` with an associated *token*. If more tokens are associated to a given location, then `PATHMAP` will have more entries for that location. The `DEPTH` attribute measures the current path length up to location identified by `LOC_ID`. `ITEM` stores information about products, in terms of EPC

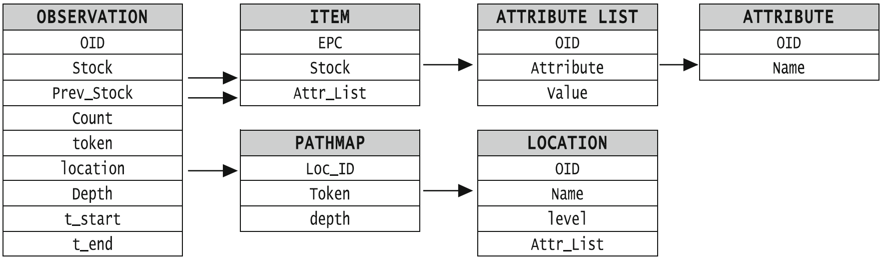


Fig. 3 Storage schema

and last stock where the item was aggregated. ATTRIBUTE gathers attributes featuring both ITEM and LOCATION and they are related to an attribute list through ATTRIBUTE_LIST which associates a value to each attribute. OBSERVATION traces all transitions of aggregates. There, the STOCK attribute is the current aggregate, containing items coming from PREV_STOCK, observed in a given LOCATION. Each observation stores stock items number as well as traversed locations (COUNT and DEPTH attributes), assigned token and time when the stock entered and left the location respectively (T_START and T_END attributes).

As resulting tables could be very large, relational DBMS capabilities have been deeply exploited tuning the above schema to obtain better performances. The first optimization is the horizontal partitioning of tables into smaller physical ones, namely *partitions*. Each partition inherits the same structure (e.g., columns and datatypes) of the original table, namely *master*. Partitioning is performed row by row and it is based on column value (*range*). For instance, in OBSERVATION table, the column LOCATION is exploited as range (see Fig. 4). If needed, the resulting smaller tables can be further partitioned with respect to another range, as for example the *time* one. Recall that this step is totally hidden from the user. STOCK, OID and LOC_ID are set as ranges for ITEM, ATTRIBUTE_LIST and PATHMAP, respectively. Moreover, tree indexes B^+ are defined on the whole set of tables. Obviously, an unclustered index will be defined for each range column used for partitioning and other indexes on the single partition. In particular, clustered indexes will be used on STOCK in each OBSERVATION partition.

5.2.3 Query Processing

The supply chain analysis is based on querying object transitions. Many approaches (Gonzalez et al. 2006) only allow to find the movement history of a given tag, by means of path selection queries, also known as *tracking queries*. Other query templates are proposed in Lee and Chung (2008) in order to enable a more flexible analysis of supply chains: besides tracking queries, *path oriented queries* are considered, divided into *path oriented retrieval queries* and *path oriented aggregate queries*. The former finds tags that satisfy given conditions about path and time, the

OBSERVATION

oid	stock	prevstock	count	token	depth	location	t_start	t_end

obs1	stock01	stock01	8	1	1	L01	2	3

obs3	stock03	stock01	2	3	2	L12	4	6

obs4	stock04	stock01	3	1	2	L11	5	7
obs6	stock06	stock01	1	1	2	L11	7	8

obs2	stock02	stock01	1	4	2	L21	4	5
obs5	stock05	stock01	1	4	2	L21	5	6
obs9	stock09	stock04	1	2	3	L21	13	16
obs10	stock10	stock06	1	2	3	L21	14	18

obs7	stock07	stock03	1	3	3	L22	7	8
obs8	stock08	stock04	2	1	3	L22	8	9

Fig. 4 Horizontal partitioning

latter compute aggregate values for tags satisfying specifications. The authors formalize query templates and express queries using an XPath-like language. Here, we will use the same formalization and we will describe a method to process tracking queries and path oriented queries. Since we use an RDBMS, queries are translated into SQL (Structured Query Language).

In a tracking query, given a tag_id, a tag flow is returned (in terms of a list of locations). There is the need to find the stock containing the item as well as the token assigned to the observation corresponding to the stock itself. In the relational schema presented above, the table `ITEM` will be queried to extract the stock_id and the `OBSERVATION` table to extract the token assigned to the related observation. Finally `PATHMAP` will be queried to return all locations annotated with the extracted token. The list of locations is ordered by the attribute `DEPTH`. The corresponding SQL code ensues.

```
SELECT P.LOC_ID
FROM PATHMAP P, OBSERVATION O, ITEM I
WHERE I.EPC=<tag_id> AND O.STOCK=I.STOCK
  AND P.TOKEN=O.TOKEN
ORDER BY P.DEPTH
```

In a path oriented retrieval query, given a path expressed in terms of an XPath-like syntax, the list of tags that followed it is returned. Also in this case both stock and token information into `OBSERVATION` are exploited to select items. Moreover, ancestor and/or parent relationships between locations have to be processed. Given two locations `l1` and `l2`, if they are in a parent-child relationship (i.e., `l1/l2`), then `PATHMAP` is queried and tokens are selected such that the depth of `l2` is the depth of `l1` incremented by 1. Otherwise, if locations are in the ancestor-descendant relationship (i.e., `l1//l2`), tokens such that the depth of `l2` is greater than the depth of `l1` are selected. Let us consider the query `//L1/L2`. In this case we translate it into SQL as reported hereafter:

```
SELECT I.EPC
FROM PATHMAP P1, PATHMAP P2, OBSERVATION O, ITEM I
WHERE P1.LOC_ID=MD5('L1') AND P2.LOC_ID=MD5('L2')
  AND P2.DEPTH=(P1.DEPTH+1) AND O.LOCATION='L2'
  AND O.TOKEN=P2.TOKEN AND I.STOCK=O.STOCK
```

When translating `//l1//l2` in SQL, the condition `P2.depth > P1.depth` has been considered. Anyway, path oriented retrieval queries may present different conditions and, in that case, selection conditions has to be added to `ITEM` and `OBSERVATION`. For instance, we can enrich the previous query with time conditions such as `//l1/l2[t_end < 200]`. With respect to the previous SQL code, condition `O.t_end < 200` must be added to the `WHERE` clause. Path oriented aggregate queries present aggregate functions which can be considered in the `SELECT` clause of the SQL query such as `COUNT`, `SUM`, `MAX` and so on.

6 Experimental Evaluation

As stated in the above Sect. 5, the proposed framework enables analyses at two different stages. In what follows, an on-line object discovery toy example is presented exploiting semantic-based object annotation in a given u-commerce context; furthermore a massive data examination conducted off-line on large amount of data is outlined. The benchmarking system is a dual quad core 2.66 GHz Intel Xeon PC, running Linux Gentoo OS, with 8 GB RAM memory, 6 MB cache memory, and a 2-disk 1Tbyte striped RAID array. We carried out our experiments using PostgreSQL 8.3, because it has been proved (see Beckmann et al. in 2006) that it is significantly more efficient with respect to commercial database tools. Two different supply chains in the field of clothing production and distribution (namely $Chain_{20}$ and $Chain_{100}$) have been defined with 20 and 100 nodes, respectively. Both chains present seven levels, i.e., in both cases a path has a maximum length equal to 7.

6.1 On-line Resource Discovery

The on-the-fly discovery basically involves the end part of the supply chain, that is customers and retailers, but it can be also applied to different stages in product manufacturing and distribution, when real-time knowledge-based decision support in needed. Our matchmaking framework, leveraging the inference services described in Sect. 4, can be adapted to a wide range of resource discovery use cases. The example scenario reported here refers to the apparel product domain. A typical supply chain interaction sequence is outlined hereafter, but for the sake of brevity only the last step will be described in detail.

1. *Selection of raw materials.* Several yarn deliveries are directed to a jacket factory. Each one is identified and described by means of semantic-enabled RFID tags attached to containers, which are scanned upon arrival at the factory warehouse. Due to traceability requirements, a tag must store quantitative data such as volume, weight and a production time stamp besides the EPC code. Qualitative information about yarn and its production process are also stored within the tag, expressed as a semantic annotation with reference to an apparel industry ontology. Information extracted via RFID readers at warehouse gate can be used for several purposes by workers equipped with handheld devices:
   - product/process requirements established in the contract with supplier can be immediately verified by means of a semantic query upon yarn characteristics;
   - each yarn type can be routed to a different warehouse area, according to specific storage requirements (e.g., temperature, humidity) (Ruta et al. 2008);
2. Manufacturing and packaging. Each yarn type can be routed from the warehouse to a different production department according to its properties. Subsequently, during product packaging operations, RFID semantic-enabled tags are written and then attached to each item before shipping. Relevant portions of the semantically annotated description of raw materials can be inherited by the final product; in our scenario, a jacket can inherit information from yarn such as type of cloth and color. It is important to note that a centralized backend infrastructure is not required for this operation: just like production takes materials as input and returns products as output, information can be transferred from tags of materials to tags of products by a mobile RFID enabled ad hoc network within the factory.
3. Sales. A customer enters an outlet, where a ubiquitous commerce system is able to assist her in discovering additional available items. Details follow.

Shop items are tagged with RFID transponders containing the EPC code, OUUID, compressed semantic description and data-oriented attributes. For our case study, we adopted a simplified apparel ontology; price and body area are used as data-oriented attributes. Body area parameter is referred to a specific part of the body, as shown in Table 5.

A request is performed on behalf of the user when she enters a dressing room to try a product on. The request is based on the semantically annotated description of the product selected by the user, which is read by an RFID interrogator installed

Table 5 Correspondence between human body and values of contextual attribute

Value	1	2	3	4	5	6
Body area	Hands	Head	Chest (outer layer)	Chest (inner layer)	Legs	Feet

into the dressing room. In order to satisfy the request, a local shop hotspot retrieves resource descriptions referred to the specified ontology from the enterprise back-end database and performs a semantic-based matchmaking. RFID readers and wireless devices deployed in the outlet cooperate through an agent-based, message-oriented middleware infrastructure. A prototype has been built upon *IBM WebSphere RFID Tracking Kit* (Chamberlain et al. 2006), which provides a general framework for the orchestration of several heterogeneous devices in mobile applications. A semantic-based layer has been integrated in order to support features specific to our approach.

Let us consider the following case study: *a man enters the outlet to buy some elegant clothing. He notices a fine dark green jacket and decides to try it on.* Sensors detect the customer entering the dressing room. The RFID reader is triggered and reads data stored within the tag attached to the jacket, then it is deactivated again. Annotated data correspond to an elegant, large-sized, dark green jacket, in mostly linen cloth, suitable for young adult men and fall season, with buttons and five pockets. The corresponding DL expression w.r.t. the reference ontology (not reported here for the sake of conciseness) is:

- $Jacket \sqcap = 1\ hasColors \sqcap \exists\ hasFastenings \sqcap \forall\ hasFastenings.Buttons \sqcap \forall\ hasMainColor.DarkGreen \sqcap \exists\ hasMaterials \sqcap \forall\ hasMainMaterial.Linen \sqcap \forall\ hasPattern.Plain \sqcap = 5\ hasPockets \sqcap \forall\ hasSize.Large \sqcap \forall\ hasSleeves.LongSleeves \sqcap \forall\ hasStyle.Elegant \sqcap \forall\ suitableForAge.YoungAdult \sqcap \forall\ suitableForGender.Male \sqcap \forall\ suitableForSeason.Fall$

The equivalent DIG representation of this fairly simple annotation is 1592 B long. Realistic object descriptions can be a few times larger. Compression brings the annotation to 210 B, so that it can be stored on the RFID tag, along with the item EPC, ontology identifier and its data-oriented parameters. In this case price is $195 and body area value is 3.

The customer is trying on the jacket, as usual in a dressing room. Meanwhile, an embedded touchscreen is activated and item details are shown. Those elements will be used by the customer for building his semantic request. Let us note that the request is a DL conjunctive query, where each concept represents a desired feature. User can customize it through a graphical user interface.

Customer likes his jacket, so he would like to look for similar ones. He sets a target price of $200 and only removes the constraint on season from the system recommendation. The DL expression for user request thus becomes:

- **D:** $Jacket \sqcap = 1\ hasColors \sqcap \exists\ hasFastenings \sqcap \forall\ hasFastenings.Buttons \sqcap \forall\ hasMainColor.DarkGreen \sqcap \exists\ hasMaterials \sqcap \forall\ hasMainMaterial.Linen \sqcap \forall\ hasPattern.Plain \sqcap = 5\ hasPockets \sqcap \forall\ hasSize.Large \sqcap \forall\ hasSleeves.LongSleeves \sqcap \forall\ hasStyle.Elegant \sqcap \forall\ suitableForAge.YoungAdult \sqcap \forall\ suitableForGender.Male$

Customer confirms his request. It is now translated into a compressed DIG description, in the same way as above, and associated with user-supplied target values for data-oriented attributes. Let us suppose the following products are available in the apparel store knowledge base:

- **S1:** an elegant, large-sized, gray suit, in mostly linen cloth, suitable for adult men and spring climate, with two button fastenings and ten pockets. Price is \$678; body area is 3:
  $Suit \sqcap \forall\, hasMainColor.LightGray \sqcap \forall\, hasMainMaterial.Cotton \sqcap =10\; hasPockets \sqcap \geq 2\, hasFastenings \sqcap \forall\, hasFastenings.Buttons \sqcap \forall\, hasPattern.Plain \sqcap \forall\, hasSize.Large \sqcap =2\; hasLegs \sqcap \forall\, hasLegs.PipeLegs \sqcap \forall\, suitableForAge.Adult \sqcap \forall\, suitableForGender.Male \sqcap \forall\, suitableForSeason.Spring \sqcap =2\; hasMaterials \sqcap \exists\, hasColors \sqcap \leq 1\, hasColors$
- **S2:** medium-sized blue jeans, suitable for casual young adult men and spring climate, with pipe legs and five pockets. Price is \$38; body area is 5:
  $Trousers \sqcap =1\; hasColors \sqcap \forall\, hasMainColor.MediumBlue \sqcap \geq 2\; hasFastenings \sqcap \leq 2\; hasFastenings \sqcap \forall\, hasLegs.PipeLegs \sqcap \forall\, hasMainMaterial.Jeans \sqcap =1\; hasMaterials \sqcap \forall\, hasPattern.Plain \sqcap =5\; hasPockets \sqcap \forall\, hasSize.Medium \sqcap \forall\, hasStyle.Casual \sqcap \forall\, suitableForAge.YoungAdult \sqcap \forall\, suitableForGender.Male \sqcap \forall\, suitableForSeason.Spring$
- **S3:** an elegant, large-sized, midnight blue jacket, in mostly linen cloth, suitable for young adult men and fall climate, with buttons and five pockets. Price is \$190; body area is 3:
  $Jacket \sqcap =1\; hasColors \sqcap \exists\, hasFastenings \sqcap \forall\, hasFastenings.Buttons \sqcap \forall\, hasMainColor.MidnightBlue \sqcap \exists\, hasMaterials \sqcap \forall\, hasMainMaterial.Linen \sqcap \forall\, hasPattern.Plain \sqcap =5\; hasPockets \sqcap \forall\, hasSize.Large \sqcap \forall\, hasSleeves.LongSleeves \sqcap \forall\, hasStyle.Elegant \sqcap \forall\, suitableForAge.YoungAdult \sqcap \forall\, suitableForGender.Male \sqcap \forall\, suitableForSeason.Fall$
- **S4:** an elegant, medium-sized striped lavender jacket, in mostly synthetic material, suitable for adult women and spring climate, with buttons and two pockets. Price is \$194; body area is 3:
  $Jacket \sqcap =3\; hasColors \sqcap \exists\, hasFacings \sqcap =1\; hasFastenings \sqcap \forall\, hasFastenings.Buttons \sqcap \forall\, hasMainColor.Lavender \sqcap =3\; hasMaterials \sqcap \forall\, hasMainMaterial.Synthetic \sqcap \forall\, hasPattern.Striped \sqcap =2\; hasPockets \sqcap \forall\, hasSize.Medium \sqcap \forall\, hasSleeves.LongSleeves \sqcap \forall\, hasStyle.Elegant \sqcap \forall\, suitableForAge.Adult \sqcap \forall\, suitableForGender.Female \sqcap \forall\, suitableForSeason.Spring$

The shop server performs the matchmaking. Results are reported in Table 6 where scores are shown in the last column; being a distance measure, a lower value implies a better match. *S3* is by far the best supply for similarity search. Among

Table 6 Matchmaking results

Supply	Compatibility (Y/N)	score(·)
S1: Men gray suit	Y	4.458
S2: Men blue jeans	N	4.618
S3: Men midnight blue jacket	Y	0.97
S4: Women lavender jacket	N	8.974

others, *S4* is incompatible with *D*, as they both represent jackets but *S4* is a women's garment.

Customer can reserve one or more items. Reservation will then be notified to the local server, so that products could be prepared in advance. *Otherwise, if customer is not satisfied with the results, he can refine his request and issue it again. Eventually he exits the dressing room to finalize his purchase.* Sensor detects the exit event and the dressing room becomes ready for another customer.

A thorough experimental evaluation of system performance is ongoing with our software-simulated RFID platform. Two kinds of experimental results can be reported so far:

1. compression rates for semantically annotated products descriptions and queries in DIG;
2. performance of reading and decoding compressed semantic resource annotations from simulated RFID tags.

Compression rate was tested with 70 DIG documents of various size (from 609 B to 793 kB). Our aim was to evaluate compression performance for both smaller instance descriptions and larger ontologies. Overall average compression rate is $92.58 \pm 3.58\%$. Higher compression rates were obviously achieved for larger documents, but even for DIG files shorter than 2 kB the result is $87.05 \pm 2.80\%$, which is surely satisfactory for our purposes.

Reading and decoding times referred to compressed semantic resource annotations from simulated RFID tags was evaluated to the aim of providing insight into the possible impact of our approach on RFID system performance. In preliminary tests using the above-mentioned RFID simulation platform, a read rate of nearly 500 tags/s was obtained, whereas independent sources estimated read rates ranging from 7 to approximately 100 tags/s with standard Class 1 Generation 2 UHF RFID systems in typical conditions (Kawakita and Mistvgi 2006). This is an early evidence that semantic-based RFID applications can have comparable performance with respect to traditional ones. The latter, in turn, will not suffer any direct performance degradation from the newly introduced features, as they will read the EPC only.

6.2 Off-line Data Analysis

Off-line batch data analysis can be performed at each node of the supply chain. Let us consider the above apparel scenario. A retail outlet can monitor sales performance by aggregating records according to attributes and performing queries in order to extract relevant information for its business goals.

Stay records stored in the outlet database contain timestamps of arrival and departure (i.e., sale) for sold items, along with EPC codes. They are aggregated by model and then by arrival time and departure time (like in the example in Sect. 5.2.1). The outlet management wants to check performance of sales of jackets in the current year. In particular, there is the need to compare sales figures among different suppliers, in order to see what jackets are most successful.

Let us suppose that the outlet is labeled by $L0$ in the supply chain location graph and that it has three suppliers $L1$, $L2$ and $L3$. In that case, for each supplier Lx, the number of sold jackets since January 1st 2010 can be retrieved through a path oriented aggregate query. It corresponds to the following expression in our adopted XPath-like language:
//Lx/L0[t_{end} >′ 2010-01-01 00:00:00′ and model=′Jacket′]

This information can be retrieved from the relational database described in the previous section by means of the following SQL query:

```
SELECT COUNT(I.EPC)
FROM PATHMAP P1, PATHMAP P2, OBSERVATION O, ITEM I,
ATTRIBUTE_LIST L, ATTRIBUTE A
WHERE P1.LOC_ID=MD5(Lx) AND P2.LOC_ID=MD5(L0)
AND P2.DEPTH=(P1.DEPTH+1) AND O.LOCATION=L0
AND O.TOKEN=P2.TOKEN AND I.STOCK=O.STOCK
AND I.T_END>'2010-01-01 00:00:00'
AND I.STOCK=L.OID AND L.ATTRIBUTE=A.OID
AND A.NAME='Jacket'
```

Note that `COUNT` aggregate operator has been used in order to evaluate sold item number. In a very similar way other interesting information can be extracted. For example, the management could be interested in knowing the average time trousers from each supplier stay in the outlet before a sale occurs, so that ordered quantities can be adjusted. It is important to point out that the selection of data aggregation attributes has a direct impact on applicable queries, therefore it must descend from KPI (Key Performance Indicators) decided by managers. Of course, the approach is general and can be applied by whatever actor in the supply chain to evaluate appropriate performance metrics with respect to its own business goals, by exploiting attribute-based data aggregation and a combination of tracking queries and path oriented queries.

An experimental campaign has been carried out to evaluate framework performance also from the massive data analysis standpoint. Here, we report main results. Performance of the proposed approach has been compared with the one provided by

Lee and Chung in (2008), from now on named *Path*. The authors devise a relational schema to store RFID data as in what follows:

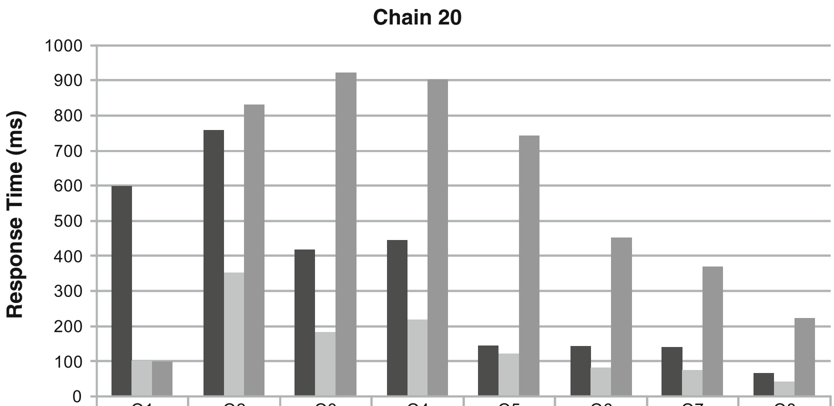
```
PATH_TABLE(PATH_ID, ELEN, OEN)
TAG_TABLE(TAG_ID, PATH_ID, START, END, TYPE)
TIME_TABLE(START, END, LOC, START_TIME, END_TIME)
INFO_TABLE(TYPE, MODEL, BRAND, CATEGORY, PRICE)
```

In this schema, `TAG_TABLE` contains information related to the trace records representation. `PATH_TABLE` stores information about path movements, according to the ELEN and OEN coefficients (recalled in Sect. 4), and `TIME_TABLE` is related to temporal information (i.e., t_{in} and t_{out} for each location). Finally, `INFO_TABLE` describes product information (e.g., model, brand, category and price). The attribute taxonomy described in Sect. 4 has been exploited synthetic stay records have been generated by setting movement and time information randomly along each chain. For both chains 5 x 10^5 stay records were created. Then two aggregations were used: $F_1 = [(t_{in} \wedge t_{out}), (EPC)]$ and $F_2 = [(t_{in} \wedge t_{out}), (model)]$. Note that F_1 corresponds to a low aggregation whereas F_2 compresses records considerably. Using templates provided in Lee and Chung (2008), we formulate eight queries. They are shown in Table 7. $Q1$ is a tracking query, $Q2$, $Q3$, $Q4$ and $Q5$ are path oriented retrieval queries and finally $Q6$, $Q7$ and $Q8$ are path oriented aggregate queries. Figure 5 shows performance test results. First of all, it has to be noticed that times in $Chain_{20}$ are higher than in $Chain_{100}$. Object transitions are more widely spread in the latter, due to a greater number of nodes.

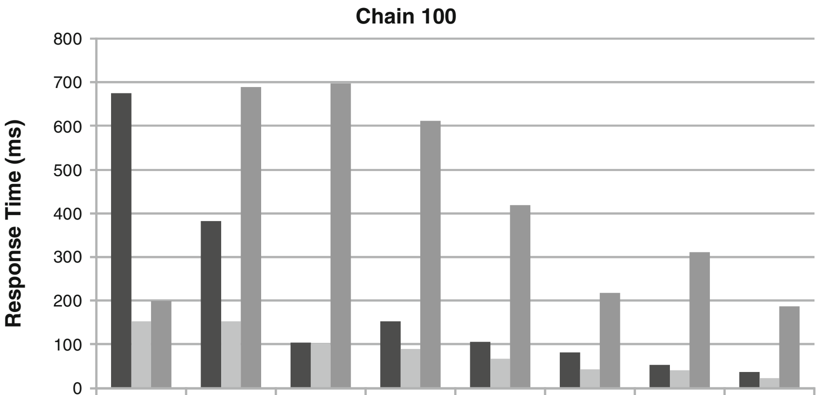
For the tracking query $Q1$, *Path* is faster than our approach with respect to F_1. However, in that system the query returns ELEN and OEN numbers, hence another application has to compute the factorization of ELEN ordering numbers with OEN.

Table 7 Test queries

Q1	7515
	17281
Q2	//L45
	//L628
Q3	/L01/L12/L22/L33/L45
	/L03/L16/L29/L321/L422/L628
Q4	//L12//L34
	//L16//L524
Q5	/L01/L12//L33/L45[t_start = 336501 AND t_end = 514515]
	/L03/L16//L321/L422/L628[t_start = 919250 AND t_end = 1023517]
Q6	COUNT()://L45
	COUNT()://L628
Q7	COUNT()://L12//L34
	COUNT()://L16//L524
Q8	COUNT():/L01/L12//L33/L45[t_start = 336501 AND t_end = 514515]
	COUNT():/L03/L16//L321/L422/L628[t_start = 919250 AND t_end = 1023517]



	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
F1	598	759	418	444	145	142	141	67
F2	101	351	183	219	123	83	75	42
Path	100	832	921	903	743	454	369	223



	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
F1	675	382	105	153	107	81	53	36
F2	153	152	101	90	68	43	40	22
Path	200	689	698	612	418	218	311	188

Fig. 5 Execution times

This is a high complexity task because the computation for large numbers is not efficient. Instead, the proposed approach directly returns the ordered list of traversed locations. Moreover performances are comparable if we use the aggregation F_2. This proves the usefulness of our aggregation mechanism.

For the path oriented retrieval queries $Q2$, $Q3$, $Q4$, $Q5$, our system exhibits better performance than *Path*. In all queries, *Path* has to execute a heavy join between `TAG_TABLE`, containing information of all items, and `PATH_TABLE`, containing ELEN and OEN numbers. Our system accesses `OBSERVATION` to extract all stocks following the input path, and then `ITEM` is accessed to return the EPC of corresponding items in those stocks. Furthermore, the partitioning technique supports the query process. Each `OBSERVATION` partition corresponds to a given location, therefore a direct access to interesting items is possible. Also in this case, the added value of aggregation is straightforward. With respect to F_1, the two approaches are comparable, whereas our system provides a better performance for the aggregation F_2. In $Q5$ time conditions lighten the processing of results. While *Path* has to introduce a new join with `TIME_TABLE`, containing timing information of transitions, our system adds a selection condition to `WHERE` clause; therefore also for the aggregation F_1 it has better performances. The path oriented aggregate queries $Q6$, $Q7$ and $Q8$ produce a similar behavior. In this case $Chain_{100}$ significantly exploits the partitioning technique, making aggregations F_1 and F_2 comparable.

7 Conclusions

This chapter presented an innovative approach for data management in supply chains based on RFID identification technology. Both *on-line* semantic-based object discovery and *off-line* analyses involving large amounts of RFID data are enabled. Distinguishing features are: (i) definite modifications to the EPCglobal standards allowing to exploit ontology-based data as well as to support non standard inference services, while keeping backward compatibility, (ii) advanced compression techniques enabling a significant space saving also maintaining a logical representation of data aggregation.

Such an approach may provide several benefits. Information about a product is structured and complete; it accurately follows the product history within the supply chain, being progressively built or updated during object lifecycle. This improves traceability of production and distribution, facilitates sales and post-sale services thanks to an advanced and selective discovery infrastructure. Indexing techniques that guarantees an efficient data access have been also proposed in a tool implementing the proposed approach.

Some experimental results are presented to show the feasibility of the proposed framework also evidencing its effectiveness. The coherent development of the approach allows a strengthening of the information to be shared between the actors involved in supply chains, reducing the costs of adoption of RFID in business. Furthermore, an increase in transparency and trust is achieved not only between supply chain partners, but also between retailers and customers. This may be a direct competitive advantage for companies that adopt the technology.

References

Angeles R (2005) RFID technologies: supply-chain applications and implementation issues. In Journal of Information System Management, **22**(1): 51–65

Ayoub Khan MBR, Manoj S (2009) A Survey of RFID Tags. In International Journal of Recent Trends in Engineering, **1**(4): 68–71

Baader F, Calvanese D, Mc Guinness D, Nardi D, Patel-Schneider P (2002) The description logic handbook. Cambridge University Press, Cambridge

Baader F, Horrocks I, Sattler U (2003) Description logics as ontology languages for the semantic web. Festschrift in honor of Jorg Siekmann, Lect Notes Artif Intell. Springer Heidelberg

Bai Y, Wang F, Liu P (2006) Efficiently filtering RFID data streams. CleanDB Workshop, 50–57, Seoul, Korea

Bai Y, Wang F, Liu P, Zaniolo C, Liu S (2007) RFID data processing with a data stream query language. Proceedings of the 23nd international conference on data engineering, ICDE 1184–1193, Istanbul, Turkey

Ban C, Hong B, Kim D (2005) Time parameterized interval r-tree for tracing tags in rfid systems. Database and Expert Systems Applications, 16th International Conference, DEXA 2005, Copenhagen, Denmark, 22-26, August 2005, Proceedings, vol. 3588 of Lecture Notes in Computer Science, pp. 503–513, Springer

Bechhofer S, Möller, R, Crowther P (2003) The DIG description logic interface. Proceedings of the 16th International workshop on description logics (DL'03), Rome, Italy, September, vol. 81 of CEUR Workshop Proceedings

Beckmann JL, Halverson A, Krishnamurthy R, Naughton JF (2006) Extending rdbmss to support sparse datasets using an interpreted attribute storage format. Proceedings of the 22nd international conference on data engineering, ICDE 2006, 3–8 April 2006, Atlanta, GA, USA, p. 58, IEEE Computer Society

Berners-Lee T, Hendler J, Lassila O (2001) The semantic Web. Sci Am **284**: 28–37

Borgida A Description logics in data management. IEEE Trans Knowledge Data Eng **7**: 671–682

Brachman R, Levesque, H (1984) The tractability of Subsumption in Frame-based description languages. 4th National Conference on Artificial Intelligence (AAAI-84), Morgan Kaufmann, Massachusetts, 34–37

Chamberlain J, Blanchard C, Burlingame S, Forestier SC, Griffith G, Mazzara M, Musti S, S.-Y, Son, Stump G, Weiss C (2006) IBM WebSphere RFID Handbook: A Solution Guide. IBM International Technical Support Organization

Colucci S, Di Noia T, Di Sciascio E, Donini F, Mongiello M (2003) Concept abduction and contraction in description logics. 16th International Workshop on Description Logics (DL'03), September, vol. 81 of CEUR Workshop Proceedings, Rome, Italy

De P, Basu K, Das S (2004) An ubiquitous architectural framework and protocol for object tracking using RFID tags. 1st International Conference on Mobile and Ubiquitous Systems:Networking and Services (MOBIQUITOUS 2004), 174–182, Boston, Massachusetts, USA

De Virgilio R, Sugamiele P, Torlone R (2009) Incremental aggregation of RFID data. Proceedings of the 2009 International Database Engineering & Applications Symposium, 194–205, ACM, Cetraro, Italy

Di Noia T, Di Sciascio E, Donini F, Mongiello M (2003) Abductive matchmaking using description logics. IJCAI '03, 337–342

Di Noia T, Di Sciascio E, Donini F, Mongiello M (2004) A system for principled matchmaking in an electronic marketplace. Int J Electron Commer, **8**: 9–37

Di Noia T, Di Sciascio E, Donini F, Ruta M, Scioscia F, Tinelli E (2008) Semantic-based Bluetooth-RFID interaction for advanced resource discovery in pervasive contexts. Int J Semant Web Inf Syst (IJSWIS), **4**: 50–74

EPCglobal Ratified Specification (October, 4, 2005) Object Naming Service (ONS - ver. 1.0). http://www.epcglobalinc.org

Gärdenfors P (1988) Knowledge in flux: modeling the dynamics of epistemic States. Bradford Books, MIT Press, Cambridge

Gonzalez H, Han J, Li X, Klabjan D (2006a) Warehousing and analyzing massive RFID data sets. Proceedings of the 22nd International Conference on Data Engineering, IEEE Computer Society Washington, DC P83

Gonzalez H, Han J, Li X, Klabjan D (2006) Warehousing and analyzing massive rfid data sets. Proceedings of the 22nd International Conference on Data Engineering, ICDE 2006, 3–8 April 2006, Atlanta, GA p. 83, IEEE Computer Society

Jeffery S, Garofalakis M, Franklin M (2006) Adaptive cleaning for RFID data streams. Proceedings of the 32nd international conference on Very large data bases, 163–174, VLDB Endowment, Seoul, Korea

Jones P, Clarke-Hill C, Shears P, Comfort D, Hillier D (2004) Radio frequency identification in the UK: opportunities and challenges. Int J Retail & Distribution Manag, **32**: 164–171

Karkkainen M (2003) Increasing efficiency in the supply chain for short shelf life goods using RFID tagging. Int J Retail Distribution Management **31**: 529–536

Kawakita Y, Mistugi J (2006) Anti-collision performance of Gen2 air protocol in random error communication link. Proceedings of the International Symposium on Applications and the Internet Workshops - SAINT 2006, 68–71, Phoenix, Arizona, USA

Lee C H, Chung C-W. (2008) Efficient storage scheme and query processing for supply chain management using rfid. Proceedings of the ACM SIGMOD international conference on management of Data, SIGMOD 2008, Vancouver, BC, Canada, 291–302 June 10–12, ACM

Li L, Horrocks I (2004) A software framework for matchmaking based on semantic web technology. Int J Elect Com **8**: 39–60

Maass W, Filler A (2006) Towards an infrastructure for semantically annotated physical products. Informatik, **94**: 544–549

Marcus DAA, Madden S, Hollenbach K (2007) Scalable semantic web data management using vertical partitioning. VLDB '07: Proceedings of the 33rd international conference on very large data bases, 411–422, VLDB Endowment, Vienna, Austria

Paolucci M, Kawamura T, Payne T, Sycara K (2002) Semantic matching of web services capabilities. The Semantic Web - ISWC 2002, pp. 333–347, no. 2342, Sardinia, Italy

Pepe MFC, Risso, M (2008) SME food suppliers versus large retailers: perspectives in the international supply chains. 15th Conference of European Institute of Retailing and Services Studies (EIRASS), 411–422, Zagreb, Croatia

Römer K, Schoch T, Mattern F, Dübendorfer T (2004) Smart identification frameworks for ubiquitous computing applications. Wire Networks, **10**: 689–700

Raza N, Bradshaw V, Hague M, (1999) Applications of RFID technology. IEE Colloquium on RFID Technology, 1–5

Ruta M, Di Noia T, Di Sciascio E, Scioscia, F, Piscitelli G (2007) If objects could talk: A novel resource discovery approach for pervasive environments. Int J Internet Protoc Technol (IJIPT) **2**: 199–217

Ruta M, Di Noia T, Di Sciascio E, Piscitelli G, Scioscia F. (2008) Semantic-based mobile registry for dynamic RFID-based logistics support. 10th International Conference on Electronic Commerce, ICEC 08, ACM Press, Pittsburgh, USA

Ruta M, Di Noia T, Di Sciascio E, Donini F (2006) Semantic-enhanced bluetooth discovery protocol for M-commerce applications. Int J Web Grid Serv **2**: 424–452

Sellitto C, Burgess S, Hawking P (2007) Information quality attributes associated with RFID-derived benefits in the retail supply chain. Int J Retail & Distrib Manag, **35**: 69–87

Smith A (2005) Exploring radio frequency identification technology and its impact on business systems. Inf Manag Compu Secur **13**: 16–28

Tajima M (2007) Strategic value of RFID in supply chain management. J Purch Manag **13**: 261–273

Traub K, et al. (2005) EPCglobal Architecture Framework. Tech. rep., EPCglobal

Wang F, Liu P (2005) Temporal management of RFID data. Proceedings of the 31st international conference on very large data bases, 1128–1139, VLDB Endowment, Trondheim, Norway

Wang F, Liu S, Liu P, Bai Y, (2006) Bridging physical and virtual worlds: complex event processing for RFID data streams. Lect Notes Comput Sci, **3896**: 588, EDBT, Munich, Germany

Weiser M (1999) The computer for the 21st century. SIGMOBILE Mob. Comput Commun Rev **3**: 3–11