Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 5 (2011) 857–864

Procedia Computer Science

The Fifth International Workshop on RFID Technology - Concepts, Applications, Challenges (IWRT 2011)

Finding Commonalities in RFID Semantic Streams

Michele Ruta[b], Simona Colucci[a,b], Floriano Scioscia[b], Eugenio Di Sciascio[b], Francesco M. Donini[a]

[a] *Università della Tuscia, Viterbo, Italy*
[b] *Politecnico di Bari, Bari, Italy*

Abstract

A stream is a time-ordered sequence of data values. It is possible to define a semantic stream as a time-ordered sequence of metadata, *i.e.*, a concept stream. It may derive from a collection of semantic annotations referred to objects/subjects whose status evolves during a process as in case of product flow in supply chains. Although a concrete added value comes from the annotation of products and processes, several issues are inherited. Particularly, concept streams typically assume a not compact form, hence a fully comprehensive concise representation is needed. Leveraging capabilities allowed by EPCglobal RFID protocol standard, the paper proposes a general framework able to provide a compact representation of large concept streams also finding informative commonalities in them so allowing automated pattern analysis and trend discovery. A case study is presented to clarify the approach.

Keywords: RFID, Semantic Web, Concept Stream

1. Introduction

Radio-Frequency IDentification (RFID) technology allows to capture and retrieve on-product information in several stages of good life-cycle so evidencing specific peculiarities of transponders as data collectors but also posing not negligible issues related to information processing and management. Information conveyed by RFIDs can be collected and analyzed as stream, *i.e.*, as time-ordered sequence of data values.

Data Stream Management Systems extend Data Base Management Systems to interrogate order-based or time-based data flows, but the huge amount and the heterogeneity of information to be evaluated makes practically unfruitful to process the overall stream. Hence, summary data structures are exploited to sum up large data blocks; they can be queried more easily but obviously return approximated results. A semantic stream is a time-ordered sequence of annotations referred to objects/subjects evolving during a process. As opposed to simplistic data streams, metadata related to both products and processes provide a semantically rich and unambiguous representation making possible to infer implicit information from the stream itself.

This paper proposes an integrated framework which exploits knowledge representation theory and languages (and in particular Description Logics (DLs) [1]) to annotate relevant events and goods so enabling added-value services. EPCglobal RFID protocol standard has been extended as in [2] and further leveraged allowing tagged objects to host machine-understandable information. Although, inevitably, concept streams have a not compact form, a fully comprehensive concise representation is so possible automatically finding trends in them. Also pattern analyses are enabled. One of the most popular inference featuring DL reasoning, *i.e.*, *Subsumption*, has been exploited in [3] to feature novel services aimed at finding commonalities in concept collections formalized in a generic DL $\mathcal{L}$.

 Selection and/or peer-review under responsibility of Prof. Elhadi Shakshuki and Prof. Muhammad Younas.
doi:10.1016/j.procs.2011.07.118

Particularly, the subsumer matrix defined in in [3] has been modified and extended here in order to represent a digest (*a.k.a.*, synopsis [4]) of a semantic stream.

The proposed framework is presented and applied in a supply chain setting. Modern chains induce high dynamism, and they are made of interactions and connections rapidly evolving and modifying. The issue of conceiving such an agile vision while taking into account objectives like total quality management, controlled optimizations and environmental impact, requires to re-design organization and structures. Information is a relevant asset for granting quality standards, to enable quick business analysis and performance evaluation in order to take corrective actions and to enable sustainability and reliability. RFID-based technology well supports chains evolution, but usually relies on a stable and fixed back-end which makes every solution only partially applicable to intrinsically volatile contexts. Furthermore, current identification mechanism –exclusively providing "true/false" replies to queries on RFID data– appears as too restrictive for advanced applications. On the contrary, given the increased storage availability (up to several kBs [5]) modern transponders provide, RFID could provide further automation of actions and processes.

The remainder of the paper is structured as in what follows. Section 3 reports on relevant related work before discussing the proposed approach in Section 4, while a case study in Section 5 highlights benefits of the proposal. Section 6 closes the paper.

2. Background

In the last decade, data stream processing systems were investigated in depth and several challenges, application scenarios and solution approaches have been proposed in the literature [4]. Most proposals are based on extensions of both data model and query semantics of traditional DBMSs (Data Base Management Systems) into *DSMSs (Data Stream Management Systems)*, in order to support *continuous queries* over an order-based or time-based data *window* that moves as new data arrive. Extensions to standard SQL or new SQL-like query languages have been proposed accordingly. The practical inability to store and process a complete stream often led to the use of summary data structures as *synopses* or *digests*. Queries over synopses return approximated results. Semantic stream processing is thus emerging as a significant research challenge and opportunity [6]. Unfortunately, in traditional knowledge-based systems, semantic-based inferences are grounded on heavyweight tools, such as temporal logic and belief revision, that are not suitable for high data volumes that change rapidly. In order to cope with these issues, the approach proposed here models semantic streams according to Description Logics (DLs) formalism and exploits specifically targeted reasoning services for stream processing. In what follows some basic background in this regard will be provided.

DLs are a family of formalisms widely employed for knowledge representation in a decidable fragment of First Order Logic. Throughout the paper, we will refer to the $\mathcal{ALN}$(D) (Attributive Language with unqualified Number restrictions and concrete Domains) DL: $\mathcal{ALN}$(D) extends the basic sublanguage of $\mathcal{ALN}$ DL with concrete features. $\mathcal{ALN}$ provides a limited set of constructs, used to describe the knowledge domain by combining the basic DL elements, namely **concept names**, representing objects of the domain –*i.e.*, $Color$, $Material$, $Size$, $Pattern$– and **role names**, referred to possible binary relationships among concepts, *i.e.*, $hasPattern$, $hasMainMaterial$, $hasPrice$, $hasProductType$. Every DL includes two special concepts, $\top$ and $\bot$, a concept interpreted by the whole domain and by an empty set, respectively. $\mathcal{ALN}$ also allows **qualified universal restrictions** –*i.e.*, $\forall hasMainMaterial.Cotton$ denotes products mainly made up by cotton– and **number restrictions** –*i.e.*, $\geq 3 hasProductType$, $\leq 2 hasProductType$ denotes objects made up by at least three or at most two product types– over roles. By extending $\mathcal{ALN}$ with concrete features, concepts can be linked to a concrete domain D (*e.g.*, integers, reals, strings and so on) through a set of unary predicates p, –*i.e.*, by $=_{50}$ $hasPrice$ we mean objects of the domain exactly pricing 50 Eur. By using such constructs it is possible to detail concept **inclusions** and **definitions**, which constitute the intensional knowledge of a DL system, called **TBox** in DL notation and **ontology** in knowledge representation. For example, the inclusion $DarkRed \sqsubseteq Red \sqcap \forall hasShade.DarkShade$ (respectively $DarkBlue \sqsubseteq Blue \sqcap \forall hasShade.DarkShade$ and $MidnightBlue \sqsubseteq Blue \sqcap \forall hasShade.DarkShade$) asserts that the set of dark red (respectively dark blue and midnight blue) objects is included in the one of red (respectively blue and blue) objects with dark shade; the concept definition $SoftGood \equiv Product \sqcap \leq_3 (hasExpirationDays)$ asserts that a soft good is a product with less than 3 days of expiration time.

The semantics of concept descriptions is conveyed through an **Interpretation** $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where $\Delta^{\mathcal{I}}$ is a non-empty set denoting the domain of $\mathcal{I}$ and $\cdot^{\mathcal{I}}$ is an interpretation function such that: i) $\cdot^{\mathcal{I}}$ maps each concept name A in a set $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$; ii) $\cdot^{\mathcal{I}}$ maps each role name R in a binary relation $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$.

3. Related work

RFID-based supply chain scenarios introduce peculiarities w.r.t. generic data stream processing, concerning both the properties of managed data and main application requirements. Consequently, several specialized solutions have been proposed, which can be divided in two broad categories. The first one concerns run-time processing of data streams [7, 8, 9, 10]. Proposed approaches bear similarities with general-purpose DSMSs. Nevertheless, they manage only very basic information, namely raw data produced by RFID readers, consisting of *(EPC, location, time)* triples, where the EPC (Electronic Product Code) is the unique product identifier and each RFID reading event is marked with location and time. The second research direction focuses on off-line computation and efficient data storage [11, 12, 13, 14]. Wang [9] 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. In [12] a location-oriented indexing was presented, tracing paths registered by RFID readers through a novel representation model. [13] and [14] provided, instead, storage models borrowed from datawarehouse literature, where multidimensional analysis is based on data aggregation along different dimensions. The main limitation is that aggregation by product characteristics is limited to trivial taxonomies of product types, which are defined in an ad-hoc fashion and lack explicit semantics.

More advanced information and knowledge representation techniques have been recently proposed for smarter supply chain management, able to support analyses with higher-level semantics. Particularly, a rich characterization of products equipped with RFID tags can be achieved by means of Semantic Web languages such as RDF[1] and OWL[2]. In [15] technological solutions were proposed to allow storage and extraction of semantically annotated product descriptions in EPCglobal UHF Generation 2 RFID tags. Semantic Web technologies allow a formalization of annotations in a machine understandable way w.r.t. shared conceptual specifications (ontologies), so promoting interoperability and sophisticated inferences at various stages of product lifecycle [16, 17]. A range of tools can be used for information processing and analysis, including rule-based systems, logic-based reasoning engines and query engines based on declarative languages such as SPARQL[3]. 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.

Several approaches have been proposed to extend standard SPARQL language and query engines for RDF knowledge bases with temporal properties. In [18], authors proposed an approach toward reasoning over streams of timestamped RDF statements, which is composed by two elements: a SPARQL extension –named C-SPARQL (Continuous SPARQL)– bringing the notion of continuous query processing (typical of DSMSs) into the language; a software system architecture with a plain open source DSMS and a plain SPARQL query engine working in pipeline. Such elements execute continuous queries over a temporal window on the RDF assertion stream. Further work [19] introduced simple reasoning in terms of incremental maintenance of materializations of ontological entailments. Similar approaches include Streaming SPARQL [20] and Time-Annotated SPARQL [21]. Streaming Knowledge Bases [22] use a SQL-like language, instead of SPARQL, and achieves high query processing scalability by preprocessing the ontology referenced by RDF statements in the stream, reformulating RDF entailment rules as SQL queries and using a tuple-based DSMS stream processor to execute them. All the above approaches [19] are limited to the simple entailment regime of RDF, therefore cannot provide the capability to detect commonalities in concept expressions that are exploited by the system proposed here to perform meaningful queries over RFID semantic streams.

[1] Resource Description Framework, W3C Recommendation 10 February 2004, http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/
[2] Web Ontology Language, version 2, W3C Recommendation 27 October 2009, http://www.w3.org/TR/owl2-overview/
[3] SPARQL Protocol And Query Language for RDF, W3C Recommendation 15 January 2008, http://www.w3.org/TR/rdf-sparql-query/

4. Theoretical framework

Before describing the theoretical framework underlying the proposed approach, the employed reasoning services will be shortly recalled in Section 4.1 to make the paper self-contained. Furthermore, the proposed semantic stream representation framework will be presented in Section 4.2.

4.1. Resource Representation and Reasoning in Description Logics

The most important –and well-known– service featuring DL reasoning checks for specificity in hierarchies, by determining whether a concept description is more specific than another one.

Definition 1 (Subsumption). *Given two concept descriptions C and D and a TBox $\mathcal{T}$ in a DL $\mathcal{L}$, we say that D subsumes C w.r.t. $\mathcal{T}$ ($C \sqsubseteq_{\mathcal{T}} D$) if for every model of $\mathcal{T}$, $C^{\mathcal{I}} \subset D^{\mathcal{I}}$. As a special case, two concepts are* equivalent *if they subsume each other.*

For example, let us consider the following concept descriptions referred to different products: $\mathbf{P_1} = Shirt \sqcap \forall hasMainColor.LightBlue$ and $\mathbf{P_2} = UpperBodyGarment \sqcap \forall hasMainColor.Blue$. Also consider the TBox $\mathcal{T}$ modeling the concept inclusions introduced in Section 2 and the one: $Shirt \sqsubseteq_{\mathcal{T}} UpperBodyGarment$. Hence, given the model, knowledge expressed by P_1 is more specific than the one required by P_2 w.r.t. $\mathcal{T}$: according to the previous definition P_2 subsumes P_1.

Based on subsumption, new reasoning services may be defined in DLs. In particular, we are interested in the ones aimed at finding commonalities in a concepts collections formalized in a generic DL $\mathcal{L}$. In what follows, several non-standard inferences significant to this aim will be recalled starting from *Least Common Subsumer* definition [23].

Definition 2 (LCS). *Let $C_1, \ldots, C_p$ be p concept descriptions in a DL $\mathcal{L}$. An LCS of $C_1, \ldots, C_p$, denoted by LCS $(C_1, \ldots$ is a concept description E in $\mathcal{L}$ s.t. the following conditions hold: i) $C_h \sqsubseteq E$ for $h = 1, \ldots, p$; ii) E is the least $\mathcal{L}$-concept description satisfying (i),* i.e.*, if E' is an $\mathcal{L}$-concept satisfying $C_i \sqsubseteq E'$ for all $i = 1, \ldots, n$, then $E \sqsubseteq E'$.*

As an example, the LCS of the above descriptions, $\mathbf{P_1}$ and $\mathbf{P_2}$, is $\mathbf{P_2}$ itself, *i.e.*, the most specific description subsuming both concepts, it represents shared features.

Several scenarios deal with the problem of identifying features shared by a significant subset of a set of concepts in DL, rather than by the set as a whole. In order to evidence such partial commonalities, specific non-standard inferences based on LCS computation have been devised [3]. In particular, common subsumers of k concepts in a collection of p elements, with $k < p$ have been defined as *k-Common Subsumers* (k-CS); furthermore, as by definition LCSs are also k-CSs, for every $k \leq p$, *Informative k-Common Subsumers* (IkCS) have been defined as a specific subset of k-CSs not subsuming all p concepts and then adding *informative* content to the LCS computation. Before continuing, such definition will be slightly refined to cope with the modeling framework presented in Section 4.2.

Definition 3 (*r*-CS, I*r*CS). *Let $C_1, \ldots, C_p$ be p concepts in a DL $\mathcal{L}$, and let be $k \leq p$. A* r-Common Subsumer *(r-CS) of $C_1, \ldots, C_p$ is a concept $D \neq \top$ such that D is an LCS of at least $r = k/p$ concepts among $C_1, \ldots, C_p$. As a special case, we define as* Informative r-Common Subsumers *(IrCS) those specific r-CSs for which $r < 1$.*

4.2. Compact Modeling of Resource Collections

Hereafter, it will be shown how to model concept collections formalized in $\mathcal{ALN}$(D) according to a compact lossless representation. Such a modeling framework allows to find commonalities in resource annotations formalized in DL. The modeling technique we propose requires concept stream elements to be written in components according to the following recursive definition:

Definition 4 (Concept Components). *Let C be a concept description in a DL $\mathcal{L}$, with C formalized as $C^1 \sqcap \cdots \sqcap C^m$. The* Concept Components *of C are defined as follows: if C^j, with $j = 1 \ldots, m$ is either a concept name, or a negated concept name, or a concrete feature or a number restriction, then C^j is a* Concept Component *of C; if $C^j = \forall R.E$, with $j = 1 \ldots, m$, then $\forall R.E^k$ is a* Concept Component *of C, for each E^k* Concept Component *of E.*

As a consequence the *Subsumers Matrix* (SM) [3] should be re-defined as in what follows.

Definition 5 (Subsumers Matrix). *Let $C_1, \ldots, C_p$ be a collection of concept descriptions C_h in a DL $\mathcal{L}$ and let $D_j \in \{D_1, \ldots, D_m\}$ be the* Concept Components *deriving from the collection. We define the* ***Subsumers Matrix*** *$T = (t_{hj})$, with $h = 1, \ldots, p$ and $j = 1, \ldots, m$ (*i.e.*, rows are in a one-to-one reference with concepts and columns are in a one-to-one reference with components), such that $t_{hj} = 1$ if the component D_j subsumes C_h, and $t_{hj} = 0$ if the component D_j does not subsume C_h.*

Basically, in a given row h, the "1" elements are at least as many as the components of C_h. They may be more since for example a component of another concept might subsume C_h as well.

Definition 6 (Concept Component Relative Cardinality ($RC^F_{D_j}$)). *. Let $C_1, \ldots, C_p$ be a concepts collection and F be a collection of concepts included in the collection itself: $F \subseteq (C_1, \ldots, C_p)$. For each concept component D_j deriving from $C_1, \ldots, C_p$, a* ***Concept Component Relative Cardinality*** *is the number of concepts in F subsumed by D_j. Such a number is $RC^F_{D_j} = \sum s_{fj}$, for each f such that $C_f \in F$.*

As proved in [2], EPCglobal RFID protocol standard can be enhanced and further leveraged allowing tagged objects to host machine-understandable information annotated in DL syntax w.r.t. a reference ontology. Homomorphic compression techniques [24] favor the practical feasibility of that. The framework proposed here exploits subsumers matrix to model RFID-based product annotations flowing through a generic supply chain by means of a compact and semantic-based representation. In particular, the specific subsumers matrix to be computed takes a set of semantic-enhanced good descriptions, formalized in $\mathcal{ALN}$(D), as input collection. By the way, in order to cope with semantic streams modeling, a more aggregated information is required. We are interested in representing a collection $S_1, \ldots, S_n$ of semantic streams, with each stream S_i made up by a collection of concept descriptions in $\mathcal{ALN}$(D) representing products in the stream. To this aim, the following, preliminary, definition of *Aggregate Collection* is needed.

Definition 7 (Aggregate Collection). *Let $S_1, \ldots, S_n$ be a collection of concept collections S_i where S_i is as a collection of p_i concept descriptions in a Description Logic $\mathcal{L}$,: $S_i = C_{1_i}, \ldots, C_{p_i}$. We define $S_1, \ldots, S_n$ an* ***Aggregate Collection****.*

For each concept component, it is relevant to map how many concepts, aggregated in every semantic stream, are subsumed by the component itself. Such a modeling step deals with the semantic-based discovery for aggregation features and trends identification. To this aim, the Definition 5 is extended by the *Aggregate Subsumers Matrix* (ASM) one.

Definition 8 (Aggregate Subsumers Matrix). *Let $S_1, \ldots, S_n$ be an aggregate collection, with $S_i = C_{1_i}, \ldots, C_{p_i}$ for $i = 1 \ldots n$. Let $D_j \in \{D_1, \ldots, D_m\}$ be the* Concept Components *deriving from all the concepts in the aggregate collection. The* ***Aggregate Subsumers Matrix*** *is defined as $A = (a_{ij})$, with $i = 1 \ldots n$ and $j = 1 \ldots m$, such that for each i, $a_{ij} = v$, with $0 \leq v \leq p_i$, where v is the number of concept descriptions in S_i subsumed by the component D_j.*

In what follows, the definition of an *Aggregate Model* for an aggregate collection $S_1, \ldots, S_n$ will be finally formalized.

Definition 9 (Aggregate Model). *Let $S_1, \ldots, S_n$ be an aggregate collection, according to Definition 7: for $i = 1 \ldots n$, S_i is a collection of concept descriptions C_{ki}, with $k = 1 \ldots p_i$,. An* ***Aggregate Model*** *for $S_1, \ldots, S_n$ is the pair $\langle T, A \rangle$, made up by the following items:*

- *T is the subsumers matrix deriving from the collection $C_1, \ldots, C_p = \bigcup(C_{ki})$, with $i = 1 \ldots n$ and $k = 1 \ldots p_i$, whose elements t_{kj} are computed through oracles to subsumption.*
- *A is the aggregate subsumers matrix deriving from the input collection $S_1, \ldots, S_n$, whose elements a_{ij} are determined by processing information stored in T. In particular, each row i in A is related to an aggregate collection S_i, defined as a collection of descriptions C_{ki} whose subsumption relationship with components deriving from $S_1, \ldots, S_n$ is stored in T. According to such a modeling, values a_{ij}, for each component D_j, are determined as Concept Component Relative Cardinality $RC^{S_i}_{D_j}$ (see Definition 6).*

When referring to ASM, the new feature defined hereafter characterizes our modeling.

Definition 10. *Referring to the ASM coming from $S_1, \ldots, S_n$, we define* Concept Component Ratio *(R_{D_j}) the number of concepts subsumed by D_j out of the collection $S_1, \ldots, S_n$ cardinality. Such a number is $R_{D_j} = \frac{\sum_{i=1}^{n} a_{ij}}{\sum_{i=1}^{n} |S_{ij}|}$.*

EPC	Timestamp	Reader ID	Arrived	Source/Dest.	Annotation	D_1	D_2	...	D_m
EPC_i	ts_i	R_i	$true/false$	$Node\_ID_i$	C_i	t_{i1}	t_{i2}	...	t_{im}

Table 1: Format of recorded semantic RFID product stream, containing the Subsumers Matrix ($i = 1, \ldots, p$)

Set	| Set |	Time	Arrived	Source/Dest.	D_1	D_2	...	D_m
S_i	$\|S_i\|$	ts_i	$true/false$	$Node\_id_i$	a_{i1}	a_{i2}	...	a_{im}

Table 2: Digest of semantic RFID stream in a supply chain node, containing the Aggregate Subsumers Matrix ($i = 1, \ldots, n$)

5. Case study

A case study is now outlined to clarify practical applications of the proposed framework. In particular, semantic RFID product flow in a warehouse is considered. Each product is described via semantic-enhanced RFID as an $\mathcal{ALN}$(D) concept expression in OWL language, encoded in a compressed format, according to the approach outlined in [16]. Feasibility and cost-effectiveness are favored by the adoption of compression algorithms and the growing availability of passive RFID tags endowed with user memory amounts in the kBs. Furthermore, storing semantically annotated product descriptions in RFID tags does not add significant performance overhead to RFID readers and data collection equipment [15]. As products arrive or depart the warehouse, they are scanned by gate RFID readers; reading events, including semantic annotations extracted from tags, are fed to a semantic DSMS which computes Concept Components and subsumption tests through a reasoning engine. An extension of the Aggregate Model, described in Section 4, is used to store both standard RFID data –EPC code, timestamp, provenance/destination– and semantic-based information. For each product, a record of Table 1 is stored in the DSMS: it can be noticed that the semantic stream so described includes the SM. In this way, analytical processing queries can combine data-oriented and logic-based criteria with greater flexibility. Since product information travels within tagged physical products themselves, each supply chain node collects data without depending on an external information infrastructure or a network with supply chain partners. Organizational complexity, technological costs and security risks are reduced.

RFID data collection is characterized by high volumes, particularly for large distribution centers that are currently at the forefront of RFID-based innovation in supply chain management. Storage and analysis of a complete semantic stream and SM can become inefficient for very small time windows. The proposed approach allows semantic-aware information aggregation in the ASM, which is used like a digest in typical DSMSs to perform massive analysis. Table 2 provides the reference structure of the ASM: each stream digest line represents a *stock* of items that have arrived or departed together (or in an arbitrarily narrow time frame). We can suppose that, in many real-world cases, products in the same stock will be rather homogeneous, but this is not a requirement of the approach. For each stock, the following information is stored: cardinality (number of individual items); timestamp (according to application requirements, either a single date/time value or a range can be stored); a flag stating whether the stock has arrived or departed; the source (respectively, destination) of the arrived (resp., departed) stock; a pointer to the full SM, if present. Different settings can be adopted, according to data volumes and application requirements. For example, a warehouse receiving/sending a daily average of 100000 products grouped in 1000 stocks could store the full SM just for the current day (up to 100000 records), an ASM aggregated by stock for the last month ($30 \times 1000 = 30000$ records), an ASM aggregated by hour for the last year ($24 \times 365 = 8760$ records for incoming products and 8760 for outgoing ones) and an ASM aggregated by day for the previous ten years ($365 \times 10 = 3650$ records for incoming products and 3650 for outgoing ones).

Let us consider a toy example of a semantically annotated product flow in a supply chain, with respect to ontology axioms reported in Section 2 and Section 4.1 (the full ontology adopted for the case study is not reported due to lack of space).

– Striped midnight blue cotton shirts (100 small items 100 medium, 100 large, all priced 60 Eur): $\mathbf{S_1} = Shirt \sqcap =_{60} hasPrice \sqcap \forall hasMainColor.MidnightBlue \sqcap \forall hasPattern.Striped \sqcap \forall hasMainMaterial.Cotton \sqcap \forall hasSize.Small$ (resp. $\forall hasSize.Medium$, $\forall hasSize.Large$)

– Striped brick red cotton shirts (50 small items, 50 medium, 100 large, all priced 50 Eur): $\mathbf{S_2} = Shirt \sqcap =_{50} hasPrice \sqcap \forall hasMainColor.BrickRed \sqcap \forall hasPattern.Striped \sqcap \forall hasMainMaterial.Cotton \sqcap \forall hasSize.Small$ (resp. $\forall hasSize.Medium$, $\forall hasSize.Large$)

– Plain light blue silk shirts (50 medium items priced 99 Eur, 50 large ones priced 109 Eur): $\mathbf{S_3} = Shirt \sqcap =_{99} hasPrice$ (resp. $=_{109} hasPrice$) $\sqcap \forall hasMainColor.LightBlue \sqcap \forall hasPattern.Plain \sqcap \forall hasMainMaterial.Silk \sqcap \forall hasSize.Medium$ (resp. $\forall hasSize.Large$)

– Dark blue wool jackets (50 small items, 50 medium, 50 large, all priced 150 Eur): $\mathbf{S_4} = Jacket \sqcap =_{150} hasPrice \sqcap \forall hasMainColor.DarkBlue \sqcap \forall hasPattern.Plain \sqcap$

Set	**| Set |**	$Shirt$	$\forall hasMain\ Material.Wool$	$\forall hasSize.\ Small$	$\forall hasSize.\ Medium$	$\forall hasSize.\ Large$	$\forall hasMainColor.\ \forall hasShade.LightShade$	$\forall hasMainColor.\ \forall hasShade.DarkShade$	…
S_1	300	300	0	100	100	100	0	300	…
S_2	200	200	0	50	50	100	0	0	…
S_3	100	100	0	0	50	50	100	0	…
S_4	150	0	150	50	50	50	0	150	…
S_5	100	0	0	0	50	50	100	0	…
S_6	150	0	150	50	50	50	0	150	…
Total	1000	600	300	250	350	400	200	600	…
CCR		0.60	0.30	0.25	0.35	0.40	0.20	0.60	…

Table 3: A subset of the columns of the Aggregate Subsumers Matrix

$\forall hasMainMaterial.Wool \sqcap \forall hasSize.Small$ (resp. $\forall hasSize.Medium$, $\forall hasSize.Large$)

– Light gray synthetic trousers (50 medium items, 50 large, all priced 90 Eur): $\mathbf{S_5} = Trousers \sqcap =_{90} hasPrice \sqcap \forall hasMainColor.LightGray \sqcap \forall hasPattern.Plain \sqcap \forall hasMainMaterial.Synthetic \sqcap \forall hasSize.Medium$ (resp. $\forall hasSize.Large$)

– Checked dark red wool sweaters (50 small items, 50 medium, 50 large, all priced 80 Eur): $\mathbf{S_6} = Sweater \sqcap =_{80} hasPrice \sqcap \forall hasMainColor.DarkRed \sqcap \forall hasPattern.Checked \sqcap \forall hasMainMaterial.Wool \sqcap \forall hasSize.Small$(resp. $\forall hasSize.Medium$, $\forall hasSize.Large$)

Table 3 reports some columns of the ASM. For reader's convenience, column totals and CCR are reported at the end of the table. It can be noticed that 1000 individual product descriptions read via RFID are summarized in just 6 records, with significant storage space reduction. Analytical processing can exploit informative commonalities in semantic descriptions of products along with time and path dimension exploited by classical RFID data management systems. In our toy example, we can answer several interesting kinds of queries.

A. Find the maximum Concept Component Ratio (CCR) in the time interval $[t_{start}, t_{end}]$.

B. Find the IrCS in the time interval $[t_{start}, t_{end}]$. This query, applied with different time frames, is useful to discover/monitor global trends. In our toy example, warehouse managers want to discover relevant product characteristics that are present in at least half the products. A query for informative common subsumers in the ASM with a threshold $r = 0.5$ (50%) will return $Shirt$ and $\forall hasMainColor.\forall hasShade.DarkShade$ concept components. This means that (i) shirts are the most relevant product and (ii) current trends favor dark colors over light ones. Adding time constraints can be useful to monitor how trends vary in time, *e.g.*, clothing color, pattern or material prevalence may vary, due to weather and fashion. This kind of semantic-based query is analoguous to find facts with a given support in data mining. Nevertheless, datawarehouse approaches [13, 14] use a simplistic concept taxonomy to classify products and perform mining, whereas our ontology-based modeling is more expressive and flexible. Furthermore, modifications to the conceptual model (*e.g.*, new product types or properties) have direct impact on the logical database schema, whereas in our case the ontology can evolve without impact on the storage structures described in Section 4.2.

C. Find the IrCS in the time interval $[t_{start}, t_{end}]$ for a given source/destination. This query can be useful to discover/monitor specific trends, *e.g.*, what are the most shipped kinds of garments by each supplier. Data slicing by source/destination is possible for any type of query.

D. Find how many items (or equivalently, the CCR) of type X arrived/departed in the time interval $[t_{start}, t_{end}]$. Semantic stream processors described in Section 3 cannot answer this kind of query due to expressiveness limitations of RDF w.r.t. OWL. This query has two sub-types.

1. If X is one of the Concept Components in the ASM, an exact answer can be computed. In our example, if we want to know how many wool garments arrived, we compute the total and CCR for the $\forall hasMainMaterial.Wool$ column.
2. If X is an arbitrary concept expression, the system is able to provide an approximated answer by analyzing the ASM. Each concept component D_j is tested for subsumption w.r.t. X, then $\sum_{i=1}^{n} min(\{a_{ij} | D_j \sqsubseteq_{\mathcal{T}} X\})$ is returned as upper bound for the number of items subsuming X. For example, let us find how many large shirts have been received: $X_1 = Shirt \sqcap \forall hasSize.Large$. Concept components in the third and sixth column of Table 3 satisfy the subsumption relationship and the result is therefore $min(\{300, 100\}) + min(\{200, 100\}) + min(\{100, 50\}) + min(\{0, 50\}) + min(\{0, 50\}) + min(\{0, 50\}) = 250$ hence $CCR(X_1) \leq 250/1000 = 0.25$. The expression can also contain concept components not present in the ASM, *e.g.*, $X_2 = Upper\_Body\_Garment \sqcap >_{100} hasPrice$ retrieves all garments for the upper part of the body with price higher than 100 Eur. In our example, all items in all stocks except S_5 subsume $Upper\_Body\_Garment$ while half the items in S_3 and all items in S_4 subsume

$>_{100}$ *hasPrice*, hence the result is $CCR(X_2) \leq (50 + 150)/1000 = 0.20$. If the full SM is available for the query time window, it can be used instead of the ASM to compute the exact answer.

6. Conclusion

Grounding on capabilities enabled by an enhanced version of EPCglobal RFID protocol standard and leveraging subsumption-based logic inferences, the paper presented a general framework for managing concept streams. The proposed approach allows to benefit from a semantically rich description of products and relevant processes involving them in order to find informative commonalities in large semantic streams. Fully automated pattern analysis and trend discovery is then possible. A case study referred to a product flow in a supply chain has been presented to clarify the proposal and to outline its benefits.

Acknowledgments

The authors acknowledge partial support of Apulia Region Strategic Project PS_025.

[1] F. Baader, D. Calvanese, D. Mc Guinness, D. Nardi, P. Patel-Schneider (Eds.), The Description Logic Handbook, Cambridge University Press, 2002.
[2] M. Ruta, T. Di Noia, E. Di Sciascio, F. Scioscia, G. Piscitelli, If objects could talk: A novel resource discovery approach for pervasive environments, International Journal of Internet Protocol Technology (IJIPT) 2 (3/4) (2007) 199–217.
[3] S. Colucci, E. Di Sciascio, F. Donini, Partial and informative common subsumers of concept collections in description logics, in: Proceedings of the Twenty First International Workshop of Description Logics, 2008.
[4] L. Golab, M. Ozsu, Data stream management issues–a survey, SIGMOD Record.
[5] B. R. Ayoub Khan M., Manoj S., A Survey of RFID Tags, International Journal of Recent Trends in Engineering 1 (4) (2009) 68–71.
[6] E. Della Valle, S. Ceri, F. van Harmelen, D. Fensel, It's a Streaming World! Reasoning upon Rapidly Changing Information, Intelligent Systems, IEEE 24 (6) (2009) 83–89.
[7] Y. Bai, F. Wang, P. Liu, Efficiently filtering RFID data streams, in: CleanDB Workshop, 2006, pp. 50–57.
[8] Y. Bai, F. Wang, P. Liu, C. Zaniolo, S. Liu, RFID data processing with a data stream query language, in: Proceedings of the 23nd International Conference on Data Engineering, ICDE, 2007, pp. 1184–1193.
[9] F. Wang, S. Liu, P. Liu, Y. Bai, Bridging physical and virtual worlds: complex event processing for RFID data streams, Lecture Notes in Computer Science 3896 (2006) 588.
[10] S. Jeffery, M. Garofalakis, M. Franklin, Adaptive cleaning for RFID data streams, in: Proceedings of the 32nd international conference on Very large data bases, VLDB Endowment, 2006, pp. 163–174.
[11] F. Wang, P. Liu, Temporal management of RFID data, in: Proceedings of the 31st international conference on Very large data bases, VLDB Endowment, 2005, pp. 1128–1139.
[12] C. Ban, B. Hong, D. Kim, Time parameterized interval R-tree for tracing tags in RFID systems, in: Database and Expert Systems Applications, Springer, 2005, pp. 503–513.
[13] H. Gonzalez, J. Han, X. Li, D. Klabjan, Warehousing and Analyzing Massive RFID Data Sets, in: Data Engineering, International Conference on, IEEE Computer Society, 2006, p. 83.
[14] B. Fazzinga, S. Flesca, E. Masciari, F. Furfaro, Efficient and effective RFID data warehousing, in: Proceedings of the 2009 International Database Engineering & Applications Symposium, ACM, 2009, pp. 251–258.
[15] T. Di Noia, E. Di Sciascio, F. Donini, M. Ruta, F. Scioscia, E. Tinelli, Semantic-based Bluetooth-RFID interaction for advanced resource discovery in pervasive contexts, International Journal on Semantic Web and Information Systems (IJSWIS) 4 (1) (2008) 50–74.
[16] R. De Virgilio, E. Di Sciascio, M. Ruta, F. Scioscia, R. Torlone, Semantic-based RFID Data Management, in: D. Ranasinghe, Q. Sheng, S. Zeadally (Eds.), Unique Radio Innovation for the 21st Century: Building Scalable and Global RFID Networks, Springer, 2010, to appear.
[17] M. Gruninger, S. Shapiro, M. Fox, H. Weppner, Combining RFID with ontologies to create smart objects, International Journal of Production Research 48 (9) (2010) 2633–2654.
[18] D. Barbieri, D. Braga, S. Ceri, E. Della Valle, M. Grossniklaus, Continuous queries and real-time analysis of social semantic data with c-sparql, in: Proceedings of the Second Social Data on the Web Workshop (SDoW 2009). CEUR Workshop Proceedings, Vol. 520.
[19] D. Barbieri, D. Braga, S. Ceri, E. Della Valle, M. Grossniklaus, Incremental reasoning on streams and rich background knowledge, The Semantic Web: Research and Applications (2010) 1–15.
[20] A. Bolles, M. Grawunder, J. Jacobi, Streaming sparql-extending sparql to process data streams, The Semantic Web: Research and Applications (2008) 448–462.
[21] A. Rodriguez, R. McGrath, Y. Liu, J. Myers, Semantic Management of Streaming Data, in: Workshop on Semantic Sensor Nets at International Semantic Web Conference, 2009.
[22] O. Walavalkar, A. Joshi, T. Finin, Y. Yesha, Streaming knowledge bases, Proc. SSWS.
[23] W. W. Cohen, H. Hirsh, Learning the CLASSIC description logics: Theorethical and experimental results, 1994, pp. 121–133.
[24] F. Scioscia, M. Ruta, Building a Semantic Web of Things: issues and perspectives in information compression, in: Semantic Web Information Management (SWIM'09). In Proceedings of the 3rd IEEE International Conference on Semantic Computing (ICSC 2009), IEEE Computer Society, 2009, pp. 589–594.