Semantic Challenges for the Variety and Velocity Dimensions of Big Data

María Bermúdez-Edo, University of Granada, Granada, Spain
Emanuele Della Valle, Politecnico di Milano, Milan, Italy
Themis Palpanas, Paris Descartes University, Paris, France

1. INTRODUCTION

With the increasing use of sensor devices, machine-to-machine communications, and social networks there are large volumes of real world data that are multi-modal, dynamic and heterogeneous. Among the main challenges of the Internet of Things, Social Media Analytics and their blending in Cyber-Social-Physical Systems is how to deal with large volumes of sensory and social data, and how to extract actionable information.

Both the academia and the industry have recently started to investigate innovative solutions to deal with the challenge of processing large-scale, multi-modal, dynamic data collected from the physical, cyber and social environments.

This special issue explores the big data solutions to extract actionable information from dynamic data coming from heterogeneous sources of information. In such environments the components of variety (heterogeneous data) and velocity (dynamic data) pose a key challenge in data federation.

The Semantic Web community, in the late 2000s, started the *Stream Reasoning* research (Della Valle, Ceri, Van Harmelen, & Fensel, 2009). Back to those days, the research on Semantic Web was focusing on the variety of data, devising data representation and processing techniques that promote integration and reasoning on available data to extract implicit information. On the other hand, the community working on event and stream processing was focusing on the velocity of data, producing systems that efficiently operate on streams of data on-the-fly according to pre-deployed processing rules or queries. Stream Reasoning explored the synergy between stream processing and reasoning (Margara, Urbani, van Harmelen, & Bal, 2014) to fully capture the requirements of modern data intensive applications (Della Valle, Dell'Aglio, Margara, 2016).

Internet of Things is considered one of the disruptive technologies that will transform our lives (Manyika, Chui, Bughin, Dobbs, & Bisson, 2013), and a technology that will have a big economic impact (Al-Fuqaha, Guizani, Mohammadi, Aledhari, & Ayyash, 2015). One of the main challenges in IoT is the predictive analytics (Li, Xu, & Zhao, 2014). Predictive analytics in IoT deals with the variety and velocity components of big data. Internet of Things platforms need to take into account the extensibility, scalability and interoperability (Li et al., 2014) of data coming from heterogeneous sources of data. In this context, semantics has been adopted as the interoperability solution (Atzori, Iera, & Morabito, 2010; Bandyopadhyay & Sen, 2011; Al-Fuqaha et al., 2015). Big data analysis should support extensibility and scalability for the variety and velocity of data.

The data base community has naturally dedicated lots of effort in the area of data stream processing. Starting from the late 1990s, the development of sensor and wireless telecommunications led to an explosion in monitoring activities, which provide continuous streams of data. Handling and querying data streams consequently attracted much attention in the databases field, and several

research prototype and commercial systems were developed (Abadi et al., 2005; Motwani et al., 2003; Luckham, 2001; Adi and Etzion, 2004). Though, the focus of these systems was on the scalability aspects, rather than on the semantics.

In the rest of this paper, we discuss in some more detail each one of the above three aspects of velocity and veracity in data streams: Section 2 describes the work in this area from the data management perspective; Section 3 presents the approaches that take into account semantics; and Section 4 focuses on solutions that have been developed for stream reasoning. Finally, we conclude in Section 5, where we briefly present the contributions of the papers accepted in this special issue.

2. DATA STREAM PROCESSING

2.1. Streaming Data

In the data management community, the research effort on data stream processing has focused on two directions. First, on the development of Data Stream Management Systems (DSMSs) (e.g., Abadi et al., 2005; Motwani et al., 2003), where the goal is to build a data management system that handles data streams as first class citizens. These systems use extensions of the SQL language in order to express queries on the data streams. The STREAM project (Motwani et al., 2003) describes a data stream management system for executing continuous queries over multiple streams. The system supports a declarative query language, and addresses problems related to query optimization, operator scheduling, and load shedding. The techniques involved in achieving the above goals are sharing the operator footprint, exploiting any streaming data constraints, and using specialized operator scheduling algorithms. Borealis (the successor of the Aurora system) is another data stream management system that has been proposed in the literature (Abadi et al., 2005), which has also led to a commercial product. It introduces an algebra for expressing the continuous queries, and deals with the problems of query optimization and scheduling operator execution, as well as load shedding.

The second direction of relevant research focused on the development of Complex Event Processing (CEP) systems. CEP is defined as "a set of tools and techniques for analysing and controlling the complex series of interrelated events that drive modern distributed information systems" (Luckham, 2001). Even though the term CEP is recently being used to describe the functionality of data stream processing systems in general, the processing engines of the first CEP systems were based on the event-condition-action paradigm, and were implemented based on technologies from rule-based systems. Considerable progress has been made in this field, with modern CEP systems being able to efficiently handle composite events (Adi and Etzion, 2004). However, several of these approaches are focused on specific domains (such as active database and network management), and are geared towards processing of declarative query and event patterns specifications.

The systems described above concentrate on the efficient support for declarative queries, and they have proven quite successful in that. At the same time though, they are constrained in that they cannot support a wide variety of more complex data analytics. Below we discuss complementary approaches for processing data streams, which require such kind of complex analytics.

2.2. Data Stream Networks

Lots of work has been done for the problem of efficient data processing in networks of data streams. The problem of evaluating queries in a sensor network is addressed by (Madden et al., 2002) and (Yao and Gehrke, 2003). Some recent studies describe an efficient, data-driven approach to the problem of continuous query answering in a network of data streams, without the need for continuous examination of the individual streams (Raza et al., 2012; Raza et al., 2015). This approach is based on models that approximate the data with quality guarantees. The goal is to then use these models to reason about the data, rather than having to examine each individual value in the data streams. This general technique results in significantly reduced communication costs. There has also been work

on discovery of frequent items (Tantono et al., 2008; Manerikar and Palpanas, 2009) and correlated/conditional frequent items (Mirylenka et al., 2013; Mirylenka et al., 2015).

These approaches are useful for certain monitoring problems (e.g., continuous monitoring of streaming data for the first statistical moments). However, it is not straightforward how to generalize them in order to efficiently support applications that require performing complex processing on the data streams. Examples of such applications are the identification of abnormal behaviour, and reasoning on the distribution of the values in the streams (Subramaniam et al., 2006).

2.3. Data Series

An important category of streaming data is that of data series (or time series). In this case, the order in which data appears in the stream has particular semantics: we are interested in the sequence of data, rather than in the individual data points. Works in the field largely revolve around the use of approximations. Several summarization techniques have been proposed in the literature and could be applied to large collections of data series (Palpanas et al., 2008). These techniques can effectively reduce the representation size and the dimensionality of the data series, and also be adapted to operate in an online fashion.

Several studies have focused on the problem of streaming data series similarity matching. One approach studied the benefits of applying summarization techniques, and developed multi-scale approximate representations of the patterns using the mean statistical measure (Lian et al., 2007). The AtomicWedgie technique identifies pre-determined patterns in a streaming data series (Wei et al., 2005). This technique uses pattern envelopes, and guarantees no false dismissals, as it is based on lower bounding matching criteria. Nevertheless, it imposes some strict requirements, namely that the compared time series are of the same size, and that they do not contain significant noise (since the similarity is measured using the rigid Euclidean measure). As a solution to the above problems, techniques based on elastic distance measures, such as Dynamic Time Warping (DTW) and Longest Common Sub Sequence (LCSS), have been developed (Sakurai et al., 2007; Marascu et al., 2012).

3. SEMANTICS ON IOT PLATFORMS

Semantics in IoT research field has focused on modelling domain knowledge of sensor networks and services. Only recently this research has been extended to cover stream sensory data (Kolozali, Bermudez-Edo, Puschmann, Ganz, & Barnaghi, 2014). In the vision of IoT as a service (which is widely accepted), there are three main concepts that should be described: entities (i.e., things), resources (i.e., devices), and services (De, Barnaghi, Bauer, & Meissner, 2011). The entities (e.g., a person moving from one room to another) are associated at each point in time with resources (e.g., a sensor in the room), and these devices offer services (e.g., temperature, or humidity readings).

3.1. Evolution of Semantics on IoT Frameworks

One of the initial efforts to model IoT data came from the Open Geospatial Consortium (OGC). OGC developed information models to describe Observation and Measurements (O&M) for sensory data. These models are part of the Sensor Web Enablement (SWE) standards (Reed, Botts, Davidson, & Percivall, 2007). Although it is a well-known taxonomy, the implementation in XML lacks semantics, unlike other description languages, such as OWL, which allow to apply description logic and to infer information. Henson et al. (2009) provide the O&M taxonomy with semantic meanings translating the XML files into OWL files (Henson, Neuhaus, & Sheth, 2009; Henson, Pschorr, & Sheth, 2009). O&M focuses on the observations of IoT, but do not define other important concepts of IoT, such as services and devices.

The most adopted ontology to model the IoT is SSN (Semantic Sensor Network Ontology) (Compton et al., 2012). SSN focuses on describing the devices (in particular the sensors) of IoT. The SSN ontology was designed to allow the interoperability of heterogeneous sensor networks. It

describes concepts such as sensors, outputs, observation values and features of interest. SSN has been adopted as the core ontology for different IoT models.

A recent European Union project, IoT-A, defines what could be seen as the European architectural model of IoT (De, Elsaleh, Barnaghi, & Meissner, 2012). This project created an information model based on SSN adding the concepts of services and entities. Other European projects have adopted IoT-A as the base information model, such as OpenIoT (Soldatos, Kefalakis, & Hauswirth, 2015) and IoT.est, which extend the IoT-A ontology with concepts and relationships to represent IoT services and tests (Wang, De, Toenjes, & Reetz, 2012).

One of the pioneering IoT platforms is Global Sensor Networks (GSN) (Aberer, Hauswirth, & Salehi, 2006). It aims to make transparent to the data consumer the underlying physical networks. It offers a federation of sensor networks by means of XML-based deployment descriptors that homogenize the sensory data. Since then, the IoT platforms have grown and as of May 2016 there existed 663 IoT platforms[11]. These platforms are either commercial, or research-oriented (Perera, Liu, & Jayawardena, 2015; Díaz, Martín, & Rubio, 2016; Mineraud, Mazhelis, Su, & Tarkoma, 2016). The main challenge now is to federate most of the data coming from this variety of platforms, and already some European projects are investigating the federation of IoT platforms, such as FIESTA-IoT[22] which uses semantics for the interoperability between testbeds, or Vital (Petrolo, Loscrì, & Mitton, 2014) which aims at federating the heterogeneous IoT platforms via semantics in a cloud-based environment with special focus on smart cities.

In this context of heterogeneous solutions, some standardization efforts are also taking place. For example, OneM2M is working in a new standard for machine to machine communications, covering aspects such as protocols, security, services, data management, etc. This initiative is also studying the possibility of adding semantics to the standard through the first draft of SAREF (Daniele, Hartog, & Roes, 2015). At its current version SAREF only covers household appliances at a physical level.

3.2. Challenges on Semantic IoT Platforms

When dealing with the variety of data, different sources of information provide not only different data types, but also different velocities of the data. Therefore, IoT platforms should be able to aggregate and federate multiple sources of information, preprocess data, and offer opportunities for extracting knowledge for the end users. These platforms have to be secure, and preserve the data privacy. In addition, they need to address quality of information issues: annotate the quality and provenance of the data, and also try to improve the quality by using combined techniques from other fields (Barnaghi, Bermudez-Edo, & Tönjes, 2015). These platforms need to deal with the variety and velocity dimensions of data by using techniques for interoperability, dynamic semantics and scalability.

One important aspect when dealing with the velocity of data is the use of lightweight ontologies that can deliver quick responses and that can adapt dynamically to the information changes (Bermudez-Edo, Elsaleh, Barnaghi, & Taylor, 2015; Bermudez-Edo, Elsaleh, Barnaghi, & Taylor, 2016). We also need efficient knowledge representation of sensory data in dynamic environments that handles real-time or almost real time annotations (Kolozali et al., 2014; Kolozali, Puschmann, Bermudez-Edo, & Barnaghi, 2016).

IoT platforms should also integrate some preprocessing of the data in the framework, limiting the amount of data to be transferred to the applications and therefore increasing the efficiency of the framework and applications.

One of the main challenges in this area is the semantic interoperability between the growing number of IoT platforms and the IoT models. As a response to this challenge, an initiative to create a network between different IoT standard boards (such as W3C, IETF. OMA, OGC, etc.) has been created, starting with a workshop that was recently organized[33]. Some of the challenges identified in this workshop are the homogeneous use of the vocabulary, homogeneity in the interactions or communication models, translation between models, the modularity and reuse of models and the runtime discovery of resources and data that can discover new elements without the need of a

predefined API (Kovatsch, Hassan, Zurich, & Hartke, 2016). The automatic code generator should be useful for developers as well, as well as the automatic translation of data. All these translations would be easier when using a REST interface.

4. STREAM REASONING

This section reviews the state of the art approaches to stream reasoning. We start from the intuition of its feasibility. We then present approaches that show how to extend the Semantic Web stack with concepts from the complex event and stream processing fields in order to obtain a RDF Stream Processing (RSP) stack. Then, we discuss how to optimize reasoning techniques to meet the reactivity requirements typical of complex event and stream processing applications, i.e., the system must the cable to produce an answer before new information arriving on the stream(s) makes such an answer obsolete. A discussion on the incomplete and noisy nature of data streams closes the section.

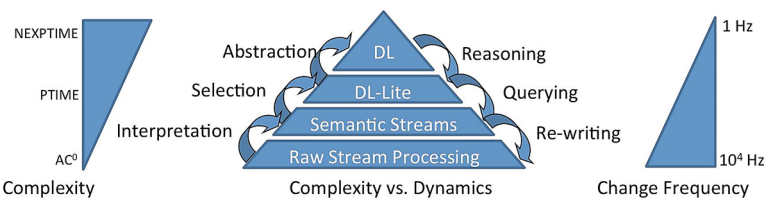
4.1. Intuition

Reasoning methods are not able to deal with high frequency data streams. This lack of reactivity is the fundamental problem of stream reasoning. While they try to derive entailments on the received data, newly incoming data can easily make those entailments obsolete. However, (Stuckenschmidt, Ceri, Della Valle, & Van Harmelen, 2010) observed that a trade-off exists between the complexity of the processing method and the frequency of the data stream a Stream Reasoner can handle. Their intuition to solve this problem is simple: as a memory hierarchy can address the trade-off between memory size and access time, a hierarchy of processing steps of increasing complexity can tackle this trade-off and allow a Stream Reasoner providing reactive answers (see Figure 1). Technically, this intuition is supported by the possibility to push processing steps down in the hierarchy to speed up reasoning, and the possibility to complete the reasoning process at each layer by only processing the results coming up from the layer underneath.

The lower layers cope with the velocity of streaming data, while the upper layers with the variety. The two bottom layers logically wrap the raw data stream into the RDF Stream data model (i.e., data is not physically mapped into RDF streams, but the layers above virtually see data as in RDF streams) and they provide the possibility to query those RDF streams using continuous extensions of the SPARQL query language under the OWL2QL entailment regime. Therefore, applying the Ontology Based Data Access (OBDA) methods, continuous queries registered on the virtual RDF streams can be rewritten in a network of continuous queries registered on the raw data streams. Only those parts of the raw stream that answer the registered queries are passed on to the higher layers, where they arrive with a lower volume/frequency.

On the next layer level, relatively simple but efficient reasoning methods, e.g., OWL2RL-based reasoning, can be used to further process the abstracted stream of results. Only at the top of the hierarchy, where the frequency of change has been reduced significantly, expressive reasoning techniques can be employed.

Figure 1. The intuition of the feasibility of stream reasoning (Stuckenschmidt, Ceri, Della Valle & Van Harmelen, 2010)



This intuition is still largely a vision; only StreamRule (Mileo, Abdelrahman, Policarpio, & Hauswirth, 2013) attempted to implement it with a two-layer approach: the first layer is a stream processing engine that acts as a filter to reduce the amount of data to be considered in the inference process. The current implementation supports the C-SPARQL Engine and CQELS, which are described in Section 4.2. The second layer is an incremental ASP (Lifschitz, 2008) reasoner that solves problems declared as the logic programs grounded in the results of the first layer to compute the answer set. As soon as the answer set is reported the solver is able to incrementally compute the next answer set based on the new data received from the first layer.

4.2. RDF Stream Processing

The stream reasoning community extended the Semantic Web stack introducing: (1) RDF stream data models to (virtually) represent data streams as a flow of RDF compliant data items, and (2) continuous querying languages based on SPARQL to continuously perform the query answering reasoning task. Systems that process RDF streams using continuous extensions of SPARQL are normally named RDF Stream Processing (RSP) Engines.

C-SPARQL (Barbieri, Braga, Ceri, Della Valle, & Grossniklaus, 2010) is a language for continuous queries over RDF streams that semantically and syntactically extends SPARQL adding operators inspired by the data stream processing model of CQL (Arasu, Babu, & Widom, 2003). The C-SPARQL engine offers a continuous execution environment for (networks of) C-SPARQL queries. It builds on top of the Esper and Jena ARQ. The engines transform each registered C-SPARQL query in a continuous query for Esper, which produces a sequence of RDF graphs over time, and a SPARQL query for Jena ARQ, which executes it against each RDF graph in the sequence and produces a continuous result. C-SPARQL offers a timestamp function to access the timestamps associated to each triple in an RDF stream. This function can be used to perform typical Complex Event Processing temporal operations (e.g., a triple appeared before another one).

CQELS-QL (Le-Phuoc, Dao-Tran, Parreira, & Hauswirth, 2011) also extends SPARQL with concepts from CQL. CQELS is the RSP engine that continuously executes *CQELS-QL* queries. Differently from the C-SPARQL engine, which offers a pluggable architecture for existing stream and SPARQL engines, CQELS evaluates queries natively. In this way, it can perform optimizations that the C-SPARQL engine cannot carry out.

$\text{SPARQL}_{\text{stream}}$ (Calbimonte, Corcho, & Gray, 2010) is another continuous extension of SPARQL. It covers all the CQL streaming operators. $\text{Morph}_{\text{stream}}$ is the execution environment of $\text{SPARQL}_{\text{stream}}$. It adopts an OBDA approach by rewriting $\text{SPARQL}_{\text{stream}}$ queries in the Event Processing Language of Esper. The only approach comparable to $\text{Morph}_{\text{stream}}$ is STARQL (Özçep, Möller, & Neuenstadt, 2014), which also adopts an OBDA approach in rewriting SPARQL queries, extended with time series operators on the Exareme stream processing engine (Kllapi, Sitaridi, Tsangaris, & Ioannidis, 2011).

These three engines cover the possible architectural variants in RSP. CQELS is a native RSP engine for RDF streams, $\text{Morph}_{\text{stream}}$ processes raw data streams with existing stream processing engines as virtual RDF streams, and the C-SPARQL engine offers a pluggable architecture that combines the benefits of existing stream processing and SPARQL engines.

4.3. Reasoning on RDF Streams

The research on Reasoning on RDF streams has both theoretical and practical dimensions.

The *theoretical* dimension addresses the need of grounding Stream Reasoning on a formal theory. To the best of our knowledge two alternative approaches are emerging (i.e., RSP-QL and LARS), and the W3C RSP community group is trying to find an agreement between them.

RSP-QL (Dell'Aglio, Della Valle, Calbimonte, & Corcho, 2014) formally models the evaluation semantics of the continuous query answering task in stream reasoning systems. RSP-QL evaluation semantics is continuous (as opposed to the one time semantics of SPARQL). Consequently, RSP-QL queries do not have *one* answer, but they have *streams of answers* computed at different time instants.

This accounts for the evolution over time of the data in the streams. The continuous semantics is the basis to introduce operators inspired by event and data stream processing, such as sliding windows and event patterns. It is worth to note that users can declare RSP-QL queries under an entailment regime of their choice; this provides for a formal definition of the stream reasoning task of continuous query answering.

LARS (Beck, Dao-Tran, Eiter, & Fink, 2015) proposes a logic to define the data model and the execution semantics of a stream reasoning engine. LARS models a stream as a sequence of time-annotated formulas. The execution semantics includes the usual logic operators (conjunction, disjunction, implication, and negation) and four temporal logic operators: $\Diamond$ indicates that a formula holds at some time in the past; $\Box$ indicated that a formula always holds in the past; $@t$ indicates that a formula holds at the specific point in time t; $\boxplus$ indicates that a formula holds in a given time interval, and is used to express the semantics of time windows. The authors prove that LARS captures the semantics of the CQL, and thus the one of the RSP query languages (i.e., C-SPARQL, CQELS-QL and $\text{SPARQL}_{\text{stream}}$) illustrated in the previous section, as well as the Etalis complex event language (Anicic, Rudolph, Fodor, & Stojanovic, 2012).

The *practical research* tackled the problem of showing the feasibility of the stream reasoning vision, illustrated in Figure 1, from two opposite directions:

1. Demonstrating that systems are capable of exhibiting a scalable performance on RDF streams, and that the materialization and incremental maintenance of ontological entailments are adequate in the streaming context;
2. Optimizing existing reasoning techniques by exploiting the natural order of the data in the streams.

The term *materialization* refers to the problem of computing all the implicit knowledge that can be derived from some given data according to some ontology. In presence of streaming data that changes frequently, techniques that maintain the materialization *incrementally* are required for efficiency.

The origin of incremental maintenance approaches can be found in maintenance of materialized views in active databases (Ceri, & Widom, 1991), where the DRed algorithm was conceived (Staudt, & Jarke, 1996); these approaches were subsequently widely used (Palpanas et al., 2002). This work considers the problem of generating a materialized view and maintaining it incrementally through a set of updates. Under certain conditions, the incremental maintenance techniques perform orders of magnitude faster than the whole re-computation of the view.

Streaming Knowledge Bases (Walavalkar, Joshi, Finin, & Yesha, 2008) is one of the earliest stream reasoning engines. Similarly, to the C-SPARQL engine it combines a stream processor (i.e., TelegraphCQ) with a reasoner (i.e., the Jena rule engine) able to incrementally materialize the knowledge base using DRed.

Ren and Pan (Ren & Pan, 2011) analyze the feasibility to optimize Truth Maintenance Systems to carry out expressive stream reasoning. Differently from Dred, they adopt a graph to track dependencies between concepts -- the nodes of the graph. New facts appearing in the stream generate new nodes and edges in the graph. When a fact become obsolete (i.e., it is retracted), they can traverse the graph and recursively remove implied facts that become unreachable.

DynamiTE (Urbani, Margara, Jacobs, Van Harmelen, & Bal, 2013) is a scalable framework to compute the materialization of a knowledge base and update it upon changes. The key novelty of the approach is the introduction of parallelization techniques to scale the system horizontally. To speed up the removal of obsolete inferences, the authors propose a novel approximate algorithm that exploits the idea of counting the number of possible ways in which a concept can be derived.

RDFox (Nenov, Piro, Motik, Horrocks, Wu, & Banerjee, J., 2015) is an in-memory RDF store. It is high scalable and it outperforms any other reasoner in its category. It uses a parallel datalog engine implementing an incremental reasoning algorithm that extends DRed. This extension reduces

the number of overestimated deletions using backward and forward reasoning to avoid the deletion of axioms that are going to be re-introduced in the rederivation step.

The four systems outlined above approach stream reasoning from the first direction illustrated in the listing above, i.e., the materialization and incremental maintenance of ontological entailments are adequate. The following five approaches belong, instead, to the second direction, i.e., the optimization of reasoning techniques by exploiting the order by recency of the data in the streams.

IMaRS (Barbieri, Braga, Ceri, Della Valle, & Grossniklaus, 2010) is an alternative to DRed for the incremental maintenance of a materialization of ontological entailments of ontologies modeled in OWL2RL. It is optimized for the stream reasoning context. It accepts only changes to the materialization caused by a window that slides over an RDF stream. It exploits the semantics of sliding windows to determine when a statement is going to expire, and marks with an expiration time all inferred data when they are deduced. This allows IMaRS to work out a new materialization by dropping explicit and inferred data whose expiration time is passed; it completely avoids the expensive step of determining which consequences become invalid. Sparkwave (Komazec, Cerri, & Fensel, 2012) implements IMaRS for RDFS on the top of the well-known Rete algorithm.

DyKnow (Heintz, & Doherty, 2004) is a middleware for autonomous agents (e.g., autonomous unmanned aerial vehicles) that sense and act in a dynamic and changing environment. Such embedded agents take as input row data from the sensors and have to create on the fly qualitative knowledge structures representing aspects of the dynamic environment where they operate. Those structures are at the basis of the qualitative reactive reasoning to perform symbol grounding, signal to symbol transformations, information fusion, contextual reasoning, and focus of attention. DyKnow uses real-time CORBA as a communication infrastructure among its distributed components.

ETALIS (Anicic, Rudolph, Fodor, & Stojanovic, 2012) is a stream reasoner able to combine complex event patter matching with RDFS reasoning. It captures event patterns as deductive rules and delegates the processing to a Prolog engine that can be plugged into the system. EP-SPARQL (Anicic, Fodor, Rudolph, & Stojanovic, 2011) is a declarative query language that extends SPARQL with typical Complex Event operators with interval bases semantics. It is possible to transform EP-SPARQL queries in rules for ETALIS. Note that while most of the systems we presented in Section 4.2 evaluate SPARQL queries without using any form of reasoning, EP-SPARQL represents an exception, since it also derives implicit knowledge before performing pattern matching to answer a query.

4.4. Incomplete and Noisy Nature of Data Streams

Streaming information is often incomplete and noisy. Existing approaches to this problem are only initial attempts in this direction. The core problems to address are similar to those that still prevent us from effectively combining inductive and deductive stream reasoning.

In the context of the analysis of social media streams, works such as (Balduini, Bozzon, Della Valle, Huang, & Houben, 2014) (Balduini, Celino, Dell'Aglio, Della Valle, Huang, Lee, ... & Tresp, 2012) (*Barbieri, Braga, Ceri, Della Valle, Huang, Tresp, ... & Wermser, 2010*) demonstrated the possibility to effectively deal with noisy and incomplete data streams by coupling deductive stream reasoning with relational learning.

Similarly, but in a more general way, (Lécué & Pan, 2013) combines statistical learning and stream reasoning to build an ontology that is used by the latter to perform reasoning. The final goal is to predict the upcoming content of the stream, e.g., the traffic conditions of cities (Tallevi-Diotallevi, Kotoulas, Foschini, Lécué, & Corradi, 2013).

(Turhan, & Zenker, 2015) suggest to extend OBDA for data streams to handle fuzzy and temporal information. The system can answer (temporal) fuzzy conjunctive queries over fuzzy data streams with respect to a (crisp) DL-Lite ontology. This allows well-known query rewriting approaches while dealing with noisy data.

(Nickles, & Mileo, 2014) proposes to use probabilistic Answer Set Programming (Baral, Gelfond, & Rushton, 2009) to process RDF data streams and Linked Data that contains potentially inconsistent information.

Probabilistic Event Calculus (Skarlatidis, Paliouras, Artikis, & Vouros, 2015) proposes to deal with uncertainty in logic-based event recognition by extending the Event Calculus (Shanahan, 1999) with probabilistic reasoning (specifically Markov logic networks) (Richardson, & Domingos, 2006).

5. IN THIS ISSUE

The papers in this special issue covered some of the challenges and open issues discussed previously.

The first article in this special issue, "Enabling RDF Stream Processing for Sensor Data Management in the Environmental Domain" (Llaves, Corcho, Taylor, & Taylor, 2016), presents a scalable extension of morph-stream (a tool for ontology based data access to data streams). Such an extension is able to cope with the variety and velocity of sensor networks. It addresses variety using the semantic sensor network (SSN) as ontology, user-defined simple mappings from CSV to SSN and conversion of data streams into "light-weight" RDF streams. At the same time, it addresses velocity using Kafka as the message queuing system, several alternative storm topologies and zookeeper.

The second article in this special issue, "Managing Large Amounts of Data Generated by a Smart City Internet of Things Deployment" (Lanza et al., 2016) shows an IoT platform for a smart city, which is developed in a city with a deployment of more than 5000 sensors or tags. The authors describe in the paper the problems and lessons learnt from the deployment of one of the biggest smart cities deployments described in the literature.

The third article in this special issue, "QoS-aware Stream Federation and Optimization based on Service Composition" (Gao, Ali, Curry, & Mileo, 2016) presents a service composition approach, selecting the services based on QoS parameters. The authors use genetic algorithms to find the optimal solution. The proposal is validated through a realistic smart city use case scenario, performing an analysis of the genetic algorithm, and a validation of the QoS aggregation rules. This article tracks an important issue dealing with the variety dimension of streaming data that is the quality of information (Barnaghi et al., 2015), and consequently, the quality of service.

María Bermúdez-Edo
Emanuele Della Valle
Themis Palpanas
Guest Editors
IJSWIS

REFERENCES

Abadi, D. J., Ahmad, Y., Balazinska, M., Çetintemel, U., Cherniack, M., Hwang, J.-H., & Zdonik, S. B. et al. (2005). *The Design of the Borealis Stream Processing Engine* (pp. 277–289). CIDR.

Aberer, K., Hauswirth, M., & Salehi, A. (2006). The Global Sensor Networks middleware for efficient and flexible deployment and interconnection of sensor networks. Retrieved from http://infoscience.epfl.ch/record/83891

Adi, A. and Etzion, O. (2004). Amit - the situation manager. The VLDB Journal 13, 2 (May. 2004), 177-203.

Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015). Internet of Things: A Survey on Enabling Technologies, Protocols and Applications. IEEE Communications Surveys & Tutorials, PP(99), 1–1. http://doi.org/<ALIGNMENT.qj></ALIGNMENT>10.1109/COMST.2015.2444095

Anicic, D., Fodor, P., Rudolph, S., & Stojanovic, N. (2011, March). EP-SPARQL: a unified language for event processing and stream reasoning. In *Proceedings of the 20th international conference on World wide web* (pp. 635-644). ACM. doi:10.1145/1963405.1963495

Anicic, D., Rudolph, S., Fodor, P., & Stojanovic, N. (2012). Stream reasoning and complex event processing in ETALIS. *Semantic Web*, *3*(4), 397–407.

Atzori, L., Iera, A., & Morabito, G. (2010). The internet of things: A survey. Computer Networks. Retrieved from http://www.sciencedirect.com/science/article/pii/S1389128610001568

Balduini, M., Bozzon, A., Della Valle, E., Huang, Y., & Houben, G. J. (2014). Recommending venues using continuous predictive social media analytics. *IEEE Internet Computing*, *18*(5), 28–35. doi:10.1109/MIC.2014.84

Balduini, M., Celino, I., Dell'Aglio, D., Della Valle, E., Huang, Y., Lee, T., & Tresp, V. et al. (2012). BOTTARI: An augmented reality mobile application to deliver personalized and location-based recommendations by continuous analysis of social media streams. *Web Semantics: Science, Services, and Agents on the World Wide Web*, *16*, 33–41. doi:10.1016/j.websem.2012.06.004

Bandyopadhyay, D., & Sen, J. (2011). Internet of things: Applications and challenges in technology and standardization. Wireless Personal Communications. Retrieved from http://link.springer.com/article/10.1007/s11277-011-0288-5

Baral, C., Gelfond, M., & Rushton, N. (2009). Probabilistic reasoning with answer sets. *Theory and Practice of Logic Programming*, *9*(01), 57–144. doi:10.1017/S1471068408003645

Barbieri, D., Braga, D., Ceri, S., Della Valle, E., Huang, Y., Tresp, V., & Wermser, H. et al. (2010). Deductive and inductive stream reasoning for semantic social media analytics. *IEEE Intelligent Systems*, *25*(6), 32–41. doi:10.1109/MIS.2010.142

Barbieri, D. F., Braga, D., Ceri, S., Della Valle, E., & Grossniklaus, M. (2010). C-SPARQL: A continuous query language for RDF data streams. *International Journal of Semantic Computing*, *4*(01), 3–25. doi:10.1142/S1793351X10000936

Barbieri, D. F., Braga, D., Ceri, S., Della Valle, E., & Grossniklaus, M. (2010, May). Incremental reasoning on streams and rich background knowledge. *Proceedings of theExtended Semantic Web Conference* (pp. 1-15). Springer Berlin Heidelberg. doi:10.1007/978-3-642-13486-9_1

Barnaghi, P., Bermudez-Edo, M., & Tönjes, R. (2015). Challenges for Quality of Data in Smart Cities. *Journal of Data and Information Quality*, *6*(2-3), 1–4. doi:10.1145/2747881

Beck, H., Dao-Tran, M., Eiter, T., & Fink, M. (2015, January). LARS: A Logic-Based Framework for Analyzing Reasoning over Streams. Proceedings of AAAI (pp. 1431-1438).

Bermudez-Edo, M., Elsaleh, T., Barnaghi, P., & Taylor, K. (2015). IoT-Lite Ontology. W3C Member Submission.

Bermudez-Edo, M., Elsaleh, T., Barnaghi, P., & Taylor, K. (2016). IoT-Lite: A Lightweight Semantic Model for the Internet of Things. Proceedings of UIC-ATC-ScalCom-CBDCom-IoP-SmartWorld.

Calbimonte, J. P., Corcho, O., & Gray, A. J. (2010, November). Enabling ontology-based access to streaming data sources. *Proceedings of theInternational Semantic Web Conference* (pp. 96-111). Springer Berlin Heidelberg. doi:10.1007/978-3-642-17746-0_7

Ceri, S., & Widom, J. (1991). Deriving production rules for incremental view maintenance.

Compton, M., Barnaghi, P., Bermudez, L., García-Castro, R., Corcho, O., Cox, S., & Taylor, K. et al. (2012). The SSN ontology of the W3C semantic sensor network incubator group. *Web Semantics: Science, Services, and Agents on the World Wide Web*, *17*, 25–32. doi:10.1016/j.websem.2012.05.003

Daniele, L., den Hartog, F., & Roes, J. (2015). Study on Semantic Assets for Smart Appliances Interoperability: D-S4: FINAL REPORT. Retrieved from http://repository.tudelft.nl/view/tno/uuid:73c44272-1ac0-4acb-a359-423d053475a6/

De, S., Barnaghi, P., Bauer, M., & Meissner, S. (2011). Service modelling for the Internet of Things. Proceedings of the 2011 Federated Conference on Computer Science and Information Systems (FedCSIS) (pp. 949–955). Retrieved from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6078180

De, S., Elsaleh, T., Barnaghi, P., & Meissner, S. (2012). An Internet of Things Platform for Real-World and Digital Objects. Scalable Computing: Practice and Experience. Doi:<ALIGNMENT.qj></ALIGNMENT>10.12694/scpe.v13i1.766

Della Valle, E., Ceri, S., Van Harmelen, F., & Fensel, D. (2009). It's a streaming world! Reasoning upon rapidly changing information. *IEEE Intelligent Systems*, *24*(6), 83–89. doi:10.1109/MIS.2009.125

Della Valle, E., Dell'Aglio, D., & Margara, A. (2016) Tutorial: Taming Velocity and Variety Simultaneously in Big Data with Stream Reasoning.*Proceedings of the 10th ACM International Conference on Distributed Event-Based Systems*. ACM. doi:10.1145/2933267.2933539

Díaz, M., Martín, C., & Rubio, B. (2016). State-of-the-art, challenges, and open issues in the integration of Internet of things and cloud computing. *Journal of Network and Computer Applications*, *67*, 99–117. doi:10.1016/j.jnca.2016.01.010

Gao, F., Ali, M. I., Curry, E., & Mileo, A. (2016). QoS-aware Stream Federation and Optimization based on Service Composition. *International Journal on Semantic Web and Information Systems*.

Heintz, F., & Doherty, P. (2004). DyKnow: An approach to middleware for knowledge processing. *Journal of Intelligent & Fuzzy Systems*, *15*(1), 3–13.

Henson, C., Neuhaus, H., & Sheth, A. (2009). An ontological representation of time series observations on the Semantic Sensor Web. Retrieved from http://corescholar.libraries.wright.edu/knoesis/676/

Henson, C., Pschorr, J., & Sheth, A. (2009). SemSOS: Semantic sensor observation service. *Proceedings of the International Symposium on Collaborative Technologies and Systems CTS '09*. Retrieved from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5067461

Kllapi, H., Sitaridi, E., Tsangaris, M. M., & Ioannidis, Y. (2011, June). Schedule optimization for data processing flows on the cloud.*Proceedings of the 2011 ACM SIGMOD International Conference on Management of data* (pp. 289-300). ACM. doi:10.1145/1989323.1989355

Kolozali, S., Bermudez-Edo, M., Puschmann, D., Ganz, F., & Barnaghi, P. (2014). A knowledge-based approach for real-time IoT data stream annotation and processing. Proceedings of the 2014 IEEE International Conference on Internet of Things (pp. 215–222). Institute of Electrical and Electronics Engineers Inc. doi:10.1109/iThings.2014.39

Kolozali, S., Puschmann, D., Bermudez-Edo, M., & Barnaghi, P. (2016). *On the Effect of Adaptive and Non-Adaptive Analysis of Time-Series Sensory Data*. IEEE Internet of Things Journal.

Komazec, S., Cerri, D., & Fensel, D. (2012, July). Sparkwave: continuous schema-enhanced pattern matching over RDF data streams.*Proceedings of the 6th ACM International Conference on Distributed Event-Based Systems* (pp. 58-68). ACM. doi:10.1145/2335484.2335491

Kovatsch, M., Hassan, Y. N., Zurich, E., & Hartke, K. (2016). *Semantic Interoperability Requires Selfdescribing Interaction Models HATEOAS for the Internet of Things*. IAB – IoT Semantic Interoperability Workshop.

Lanza, J., Sotres, P., Sanchez, L., Galache, J. A., Santana, J. R., Gutiérrez, V., & Muñoz, L. (2016). Managing Large Amount of Data Generated by a Smart City Internet of Things Deployment. *International Journal on Semantic Web and Information Systems*.

Le-Phuoc, D., Dao-Tran, M., Parreira, J. X., & Hauswirth, M. (2011, October). A native and adaptive approach for unified processing of linked streams and linked data. *Proceedings of theInternational Semantic Web Conference* (pp. 370-388). Springer Berlin Heidelberg. doi:10.1007/978-3-642-25073-6_24

Lécué, F., & Pan, J. Z. (2013, August). Predicting Knowledge in an Ontology Stream. In IJCAI.

Li, S., Da Xu, L., & Zhao, S. (2014). The internet of things: A survey. *Information Systems Frontiers*, *17*(2), 243–259. doi:10.1007/s10796-014-9492-7

Lian X., Chen L., Yu J. X., Wang G., Yu G. (2007). Similarity Match Over High Speed Time-Series Streams. *ICDE*.

Lifschitz, V. (2008, July). What Is Answer Set Programming? In AAAI (Vol. 8, pp. 1594-1597).

Llaves, A., Corcho, O., Taylor, P., & Taylor, K. (2016). Enabling RDF Stream Processing for Sensor Data Management in the Environmental Domain. *International Journal on Semantic Web and Information Systems*.

Luckham, D. C. (2001). *The Power of Events: an Introduction to Complex Event Processing in Distributed Enterprise Systems*. Addison-Wesley Longman Publishing Co., Inc.

Madden, S., Franklin, M. J., & Hellerstein, J. M. (2002). TAG: A Tiny Aggregation Service for Ad-Hoc Sensor Networks. In OSDI. doi:10.1145/1060289.1060303

Manerikar, N., & Palpanas, T. (2009). Frequent items in streaming data: An experimental evaluation of the state-of-the-art. *Data & Knowledge Engineering*, *68*(4), 415–430. doi:10.1016/j.datak.2008.11.001

Manyika, J., Chui, M., Bughin, J., Dobbs, R., & Bisson, P. (2013). Disruptive technologies: Advances that will transform life, business, and the global economy. Retrieved from http://chrysalix.com/pdfs/mckinsey_may2013.pdf

Marascu A., Khan S. A., Palpanas T. (2012). Scalable Similarity Matching in Streaming Time Series. In *PAKDD* (Vol. 2, pp. 218-230).

Margara, A., Urbani, J., van Harmelen, F., & Bal, H. (2014). Streaming the web: Reasoning over dynamic data. *Web Semantics: Science, Services, and Agents on the World Wide Web*, *25*, 24–44. doi:10.1016/j.websem.2014.02.001

Mileo, A., Abdelrahman, A., Policarpio, S., & Hauswirth, M. (2013, July). Streamrule: a nonmonotonic stream reasoning system for the semantic web. *Proceedings of theInternational Conference on Web Reasoning and Rule Systems* (pp. 247-252). Springer Berlin Heidelberg. doi:10.1007/978-3-642-39666-3_23

Mineraud, J., Mazhelis, O., Su, X., & Tarkoma, S. (2016). A gap analysis of Internet-of-Things platforms. *Computers & Society*. doi:10.1016/j.comcom.2016.03.015

Mirylenka K., Cormode G., Palpanas T., Srivastava D. (2013). Finding interesting correlations with conditional heavy hitters. ICDE: 1069-1080

Mirylenka, K., Cormode, G., Palpanas, T., & Srivastava, D. (2015). Conditional heavy hitters: Detecting interesting correlations in data streams. *The VLDB Journal*, *24*(3), 395–414. doi:10.1007/s00778-015-0382-5

Motwani, R., Widom, J., Arasu, A., Babcock, B., Babu, S., Datar, M., & Varma, R. et al. (2003). *Query Processing, Approximation, and Resource Management in a Data Stream Management System*. CIDR.

Nenov, Y., Piro, R., Motik, B., Horrocks, I., Wu, Z., & Banerjee, J. (2015, October). RDFox: A highly-scalable RDF store. *Proceedings of theInternational Semantic Web Conference* (pp. 3-20). Springer International Publishing.

Nickles, M., & Mileo, A. (2014, September). Web stream reasoning using probabilistic answer set programming. *Proceedings of theInternational Conference on Web Reasoning and Rule Systems* (pp. 197-205). Springer International Publishing. doi:10.1007/978-3-319-11113-1_16

Özçep, Ö. L., Möller, R., & Neuenstadt, C. (2014, September). A stream-temporal query language for ontology based data access. *Proceedings of the Joint German/Austrian Conference on Artificial Intelligence (Künstliche Intelligenz)* (pp. 183-194). Springer International Publishing.

Palpanas, T. (2002). Incremental Maintenance for Non-Distributive Aggregate Functions. In VLDB (pp. 802–813).

Palpanas, T., Vlachos, M., Keogh, E. J., & Gunopulos, D. (2008). Streaming Time Series Summarization Using User-Defined Amnesic Functions. *IEEE Transactions on Knowledge and Data Engineering*, *20*(7), 992–1006. doi:10.1109/TKDE.2007.190737

Perera, C., Liu, C. H., & Jayawardena, S. (2015). The emerging internet of things marketplace from an industrial perspective: a survey. *IEEE Transactions on Emerging Topics in Computing*, PP(99), 13. http://doi.org/.<ALIGNMENT.qj></ALIGNMENT>10.1109/TETC.2015.2390034

Petrolo, R., Loscrì, V., & Mitton, N. (2014). Towards a smart city based on cloud of things. *Proceedings of the 2014 ACM international workshop on Wireless and mobile technologies for smart cities* (pp. 61-66). Retrieved from http://dl.acm.org/citation.cfm?id=2633667

Raza U., Camerra A., Murphy A. L., Palpanas T., Picco G. P. (2012). What does model-driven data acquisition really achieve in wireless sensor networks? *Proceedings of PerCom* (pp. 85-94).

Raza, U., Camerra, A., Murphy, A. L., Palpanas, T., & Picco, G. P. (2015). Practical Data Prediction for Real-World Wireless Sensor Networks. *IEEE Transactions on Knowledge and Data Engineering*, *27*(8), 2231–2244. doi:10.1109/TKDE.2015.2411594

Reed, C., Botts, M., Davidson, J., & Percivall, G. (2007). Ogc® sensor web enablement: overview and high level architecture. *Proceedings of 2007 IEEE Autotestcon*. Retrieved from https://www.infona.pl/resource/bwmeta1.element.ieee-art-000004374243

Ren, Y., & Pan, J. Z. (2011, October). Optimising ontology stream reasoning with truth maintenance system. *Proceedings of the 20th ACM international conference on Information and knowledge management* (pp. 831-836). ACM. doi:10.1145/2063576.2063696

Richardson, M., & Domingos, P. (2006). Markov logic networks. *Machine Learning*, *62*(1-2), 107–136. doi:10.1007/s10994-006-5833-1

Sakurai Y., Faloutsos C., Yamamuro M. (2007). Stream Monitoring under the Time Warping Distance. In *ICDE* (pp. 1046-1055).

Shanahan, M. (1999). The event calculus explained. In *Artificial intelligence today* (pp. 409–430). Springer Berlin Heidelberg. doi:10.1007/3-540-48317-9_17

Sharmila Subramaniam et al. (2006). Online Outlier Detection in Sensor Data Using Non-Parametric Models. In VLDB (pp. 187–198).

Skarlatidis, A., Paliouras, G., Artikis, A., & Vouros, G. A. (2015). Probabilistic event calculus for event recognition. *ACM Transactions on Computational Logic*, *16*(2), 11. doi:10.1145/2699916

Soldatos, J., Kefalakis, N., & Hauswirth, M. (2015). Openiot: Open source internet-of-things in the cloud. In *Interoperability and Open-Source Solutions for the Internet of Things* (pp. 13-25). Retrieved from http://link.springer.com/chapter/10.1007/978-3-319-16546-2_3

Staudt, M., & Jarke, M. (1996, September). Incremental maintenance of externally materialized views. In VLDB (Vol. 96, pp. 3-6).

Stuckenschmidt, H., Ceri, S., Della Valle, E., & Van Harmelen, F. (2010). Towards expressive stream reasoning. In *Dagstuhl Seminar Proceedings 10042 (Semantic Challenges in Sensor Networks, 24.01. - 29.01.2010). Schloss Dagstuhl-Leibniz-Zentrum für Informatik*.

Tallevi-Diotallevi, S., Kotoulas, S., Foschini, L., Lécué, F., & Corradi, A. (2013, October). Real-time urban monitoring in dublin using semantic and stream technologies. *Proceedings of theInternational Semantic Web Conference* (pp. 178-194). Springer Berlin Heidelberg. doi:10.1007/978-3-642-41338-4_12

Tantono F. I., Manerikar N., Palpanas T. (2008). Efficiently Discovering Recent Frequent Items in Data Streams. In *SSDBM* (pp. 222-239).

Turhan, A. Y., & Zenker, E. (2015). Towards temporal fuzzy query answering on stream-based data. *Proceedings of the 1st Workshop on High-Level Declarative Stream Processing co-located with the 38th German AI conference (KI 2015)*, Dresden.

Urbani, J., Margara, A., Jacobs, C., Van Harmelen, F., & Bal, H. (2013, October). Dynamite: Parallel materialization of dynamic rdf data. *Proceedings of theInternational Semantic Web Conference* (pp. 657-672). Springer Berlin Heidelberg. doi:10.1007/978-3-642-41335-3_41

Walavalkar, O., Joshi, A., Finin, T., & Yesha, Y. (2008, October). Streaming knowledge bases.*Proceedings of the International Workshop on Scalable Semantic Web Knowledge Base Systems*.

Wang, W., De, S., Toenjes, R., & Reetz, E. (2012). A comprehensive ontology for knowledge representation in the internet of things. Trust, Security and. Retrieved from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6296201

Wei L., Keogh E. J., Van Herle H., Mafra-Neto A. (2005). Atomic Wedgie: Efficient Query Filtering for Streaming Times Series. In *ICDM* (pp. 490-497).

Yao, Y., & Gehrke, J. (2003). Asilomar, CA, USA: Query Processing for Sensor Networks. In CIDR.

ENDNOTES

1 Srđan Krčo, CEO, DunavNET in the opening of the IoT week, 2016.
2 http://fiesta-iot.eu/
3 https://www.iab.org/activities/workshops/iotsi/