



HAL
open science

An Approach for Real-time Stream Reasoning for the Internet of Things

Markus Endler, Jean-Pierre Briot, Francisco Silva E Silva, Vitor P de Almeida, Edward H Haeusler

► **To cite this version:**

Markus Endler, Jean-Pierre Briot, Francisco Silva E Silva, Vitor P de Almeida, Edward H Haeusler. An Approach for Real-time Stream Reasoning for the Internet of Things. 1st International Workshop on Semantic Multimedia Computing (SMC'17), 11th IEEE International Conference on Semantic Computing (ICSC'2017), IEEE, Jan 2017, San Diego, Californie, United States. pp.348-353. hal-01422474v1

HAL Id: hal-01422474

<https://hal.sorbonne-universite.fr/hal-01422474v1>

Submitted on 26 Dec 2016 (v1), last revised 5 Mar 2017 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

An Approach for Real-time Stream Reasoning for the Internet of Things

Markus Endler
Department of Informatics,
Pontifícia Universidade Católica
do Rio de Janeiro, Brazil
endler@inf.puc-rio.br

Jean-Pierre Briot
Sorbonne Universités, UPMC Univ Paris 06,
CNRS, LIP6 UMR 7606, France
Department of Informatics,
Pontifícia Universidade Católica
do Rio de Janeiro, Brazil
jean-pierre.briot@lip6.fr

Francisco Silva e Silva
LSDi,
Universidade Federal do Maranhão,
Brazil
fssilva@lsdi.ufma.br

Vitor P. de Almeida
Department of Informatics,
Pontifícia Universidade Católica
do Rio de Janeiro, Brazil
valmeida@inf.puc-rio.br

Edward H. Haeusler
Department of Informatics,
Pontifícia Universidade Católica
do Rio de Janeiro, Brazil
hermann@inf.puc-rio.br

Abstract—As distributed IoT applications become larger and more complex, the simple processing of raw sensor and actuation data streams becomes impractical. Instead, data streams must be fused into tangible facts and these pieces of information must be combined with a background knowledge to infer new bits of knowledge. And since many IoT applications require almost real-time reactivity to stimuli from the environment this information inference process has to be performed in a continuous, on-line manner. This paper proposes a new semantic model for data stream processing and real-time symbolic reasoning based on the concepts of *Semantic Stream* and *Fact Stream*, as a natural extensions of Complex Event Processing (CEP) and RDF (graph-based knowledge model). The main advantages of our approach are that: (a) it considers time as a key relation between pieces of information; (b) the processing of streams can be implemented using CEP and that (c) it is general enough to be applied to any Data Stream Management System (DSMS).

Index Terms—Internet of Things (IoT); sensors; data streams; complex event processing (CEP); semantic reasoning.

I. INTRODUCTION

Several complex IoT applications, such as of manufacturing industry, transportation systems, and healthcare, put hard real time requirements on the acquisition and processing of sensor data for identifying situations and extracting information from systems' operations and its environment. These typically require on-line processing of continuous streams of sensor data (Data Stream Processing), sensor fusion techniques, pattern recognition and timely and autonomous systems control. There are already many well-established machine learning algorithms for clustering, classification and sequence-analysis.

However, so far however in current IoT systems, sensing and actuation is mostly done at the *bare bones* data level, whereas many IoT applications demand higher level situation awareness of – and reasoning about – the systems' states and the physical environment where they operate. For this to be

possible, it is necessary to have comprehensive semantic models for data stream analysis and actuation. Semantic models are formally defined concepts and relations on which reasoning engines can operate to derive new bits of information and knowledge about a system and its environment. The main problem is that current semantic models (mostly designed for from the Semantic Web) are not suitable for efficient continuous-mode reasoning. Current data analysis for IoT systems is either done off-line or lacks any semantic-based reasoning.

For example, consider a production plant in near future, where several – mobile or stationary – robots operate in a product assembly and interact with each other to hand over parts and tools of the assembly line. Suddenly, there is a short power outage and the assembly line stops for a few seconds, so that some robots go back to their consistent initial states, while others continue their activity (e.g. on battery power) and only stop when their sensors notice that the production line is not advancing. In this case, the robots have to “understand” what has happened, and have to “know” which of the machinery (and robots) are in which state when activity is resumed, as well as the assembly stage of items being produced. And like magic, only a few seconds after energy is back, the robots synchronize with each other, identify missed steps in the assembly process of each item, and resume cooperating again. Such knowledge and understanding is only possible because all robots have not only a semantic model of their own state, but also situational awareness, i.e. a comprehensive model of the production process as a whole and their role in the entire process. The semantic model furthermore describes possible localized and global problems of the entire production process, as well as individual and specific actuation plans for some situations. As all possible situations cannot be represented

in a model, the robots have to classify features, combine situational patterns and combine parts of specific action plans. In the aforementioned IoT scenario, the robots would be capable of such fast recovery of the manufacturing process because their situational understanding (i.e. semantic-centered inference/reasoning process) is executed very fast, with almost no delay, as soon as each robot's operational capability is back.

With the goal of finding a suitable semantic model for IoT, this paper proposes a novel approach for real-time symbolic reasoning based on the concepts of *Semantic Stream* and *Fact Stream*, as natural extensions of Complex Event Processing (CEP) [1] and RDF (graph-based knowledge model) [2]. The main advantages of our approach are that: (a) it considers time marks as the link between simple events/pieces of information; (b) the processing of semantic streams can be implemented using existing CEP technology and that (c) it is generic enough to be applied to many Data Stream Management Systems (DSMS). This research is being carried out in the scope of the ESMOCYP research cooperation project between Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio), Universidade Federal do Maranhão (UFMA) and University of Stuttgart.

The paper is structured as follows. In Section II, we explain the basic concepts of Complex Event Processing (CEP) and list some common approaches for modeling knowledge and performing reasoning. Section III explains the two steps of semantic stream reasoning. In Section IV, we present a scenario to explain how our reasoning process would be performed using temperature and accelerometer sensors embedded into vehicles, houses and in the street. Then we present related work and in Section VI discuss the benefits of our approach and prospects. Section VII then concludes the paper.

II. FUNDAMENTALS

A. Complex Event Processing

Complex Event Processing (CEP) [1] provides a rich set of concepts and operators for processing events, which include the CQL-like (Continuous Query Language) [3], [4] queries, rules, primitive functions (aggregation, filtering, transformation, etc.) and production of derived events. A CEP workflow continuously processes incoming events, analyzes and manipulates them, and outputs derived events that are delivered to event consumers (e.g. monitoring applications). These output usually represent notifications about detected situations of interest to the the applications.

The manipulations of events are described by CEP rules, which are Event-Condition-Actions that combine continuous query primitives with context operators (e.g. temporal, logical, quantifiers) on received events, checking for correlations among these events, and generating complex (or composite) events that summarize the correlation of the input events. For example, a split rule takes an input event and creates a set of events, while a filter rule only outputs events that satisfy a given criteria. Rules can also operate on a collection of events, for example, an aggregate rule outputs a single event by executing a function on the grouped events, while

a join transformation tries to correlate events from various data streams. Another important concept in CEP is that of sliding time and event windows [4]. A time window is a temporal context that subdivides the stream of events into intervals, where CEP rules and operators are applied only to the events within each window. CEP supports three sorts of windows, Landmark, Sliding, and Fading., the latter being a sliding window where a decay factor λ is applied to the events according to their age. Most CEP systems have the concept of Event Processing Agents (EPAs), which are software modules that implement one transformation within the event processing workflow. The type of an EPA is defined by the rules it implements, such as filtering, counting or specific event pattern detection.

B. Knowledge Representation and Reasoning Approaches

There are plenty of Semantic Models that represent knowledge about a system and its environment, but almost all of them have problems of scale (i.e. the reasoning has high computational complexity) and thus are not suitable for real-time reasoning. The main semantic approaches are:

- **Frame Based Models:** A frame is an artificial intelligence data structure used to divide knowledge into substructures by representing "stereotyped situations". They are used in Artificial Intelligence Frame-based representation languages.
- **Conceptual Graphs:** Are a logical formalism that includes classes, relations, individuals and quantifiers. This formalism is based on semantic networks, but it has direct translation to the language of first order predicate logic, from which it takes its semantics.
- **Description Logic:** Are logics serving primarily for formal description of concepts and roles (relations). These logics were created from the attempts to formalize semantic networks and frame based systems. Semantically they are based on predicate logic.
- **Ontologies:** An ontology is a semantic/concept network that contains a body of knowledge describing some domain, typically common sense knowledge relating concepts.
- **Semantic Web: RDF, RDFS and OWL:** RDF is a framework for representing information about resources in a graph model, where information is represented by triples subject-predicate-object. RDFS extends RDF vocabulary to allow describing taxonomies of classes and properties. It also extends definitions for some of the elements of RDF. For example it sets the domain and range of properties and relates the RDF classes and properties into taxonomies using the RDFS vocabulary. Web Ontology Language (OWL) brings the expressive and reasoning power of Description Logic (DL) to the Semantic Web. It is divided into two levels: OWL Lite and OWL DL, which differ in their expressive power and the deduction complexity. The problem with OWL Lite and OWL DL is that reasoning is hardly implemented in an efficient way, and it also suffers from lack of scalability.

III. GENERAL IDEA

The general idea of our semantic model and reasoning approach is to define two-level CEP transformations, each of which transforms one event flow/stream into a semantically richer one: at the first stage, we transform the stream of data events (attributes are data items) into a stream of RDF statements, and at the second stage, we transform the stream of RDF-triples into semantically richer facts, i.e. combining RDF statements. The details of each step are explained in the following.

A. Mapping Data Events to Semantic Events

Our reasoning approach dictates that each simple event (i.e., an object with attributes) represents an action-based *predicate* (i.e. the event is the outcome of an action) and has at least one of the other two remaining RDF elements: the *subject* or the *object*. If the event has the ID of the subject and the object then we have the complete RDF triple (*Subject, Predicate, Object*), but otherwise, the missing third RDF element of the triple may be inferred from the shared context (i.e. the temporal and spatial correlation) of both elements, the subject and the object when these are received in separate events. For example, if we consider RDF statement (ball, kicking, in the front-yard), then the event instances represent the predicate kick. It further carries the ID of either the ball (e.g. when the ball carries an accelerometer sensor), or else the ID of the yard (e.g. the GPS-position or the street number of the yard where lawn sensors detect some kicking object). The shared context is defined by the co-location and the synchronicity of the events (detected by sensors on the yard ground and by the sensor in the ball). This contextual correlation is performed by CEP rules called “Context mappers”, that analyze the streams of events and match Subjects, Objects and Predicates.

Figure 1 shows how Context mappers analyze each pair of events in the sliding time window (e.g. 60 s.) of Data Event Stream and try to identify common contexts, based on time proximity or any other data attribute.

B. Mapping Semantic Events to Knowledge Facts

The mapping from Semantic Events (i.e. RDF triples) to Facts is achieved by Semantic Event (SEv) rules. These are CEP rules that look to find causality and temporal patterns in several Semantic Event sub-Streams, where each stream comprises the Semantic events of a given context. This “context-specific splitting” is possible in most CEP engines by the concept of a stream partition (a.k.a. context). Then, depending on the SEv rule, it might consume, filter, modify or even insert new RDF triples in some SEv streams, a feature that is supported by CEP. This manipulation is achieved by querying the Knowledge base about all the concepts and relations pertaining to the sub-streams analyzed. For example, the inference might deduce that the kicking ball with a given ID has Bob as its owner, and that the yard where the ball is kicking is the one where Bob lives. By this, the new piece of knowledge may be derived

such as someone is kicking Bob’s ball on his house’s yard. And maybe with the context information Bob has finished his homework, it is possible to deduce – with high probability – that Bob is kicking his ball in his house’s yard.

The Knowledge base is organized as *nested contexts* [5], which allows a much more efficient checking of concepts and relations when compared to single-layer (or flat) ontologies. For example, the ontology of the Knowledge Base may be organized as following nested contexts.

- Spatial nested contexts: Green Way district \supset house at 10 Rodeo Dr. \supset its yard \supset its lawn;
- Temporal nested contexts: Bob’s leisure time \supset Thursday \supset afternoon \supset Bob’s homework finished;
- Containment nested context, such as, Bob’s toys \supset Balls \supset basket ball with ID, etc.

Figure 2 shows how Semantic Event rules analyze all RDF triples in the sliding time window (e.g. 180 s.) of sub streams of Semantic Events, trying to find event patterns, filtering, manipulating or adding RDF triples into “their” main context sub-stream or also of sub-streams of semantically related contexts, such as, the front yard and the street in front of the yard.

C. Deriving Situations

Using the Facts of the stream and checking them against the Semantic Graph (Ontology) of the Knowledge base, complex situations may be identified such as Bob is playing basketball in the front yard, but should be notified that a strong storm is approaching his house’s yard. Moreover, some of the complex facts may be used for expanding, reinforcing or removing some the knowledge about a subject, an object or a place. For example, after Bob’s pen has finishes writing QED on the page with the exercises of his Math’s homework notebook, the latter has been closed, and Bob house’s main door has been opened and closed, sensing that someone left the house, then the Knowledge Base will be expanded with the following facts: (Bob, finished, Math homework), (Bob, left, house) and (Bob, stepped into, yard).

IV. AN EXAMPLE OF REASONING OVER DATA STREAMS

In this section we show how the aforementioned two-phase reasoning could be done with off-the-shelf components and current wireless WPAN technologies, such as Bluetooth Low Energy (BLE). Consider a scenario where smart ambient sensors are everywhere: in houses, offices, public transportation, in the streets and in private cars, and that these smart devices include a temperature and an accelerometer sensor, have a unique UUID and Bluetooth Low Energy interface. Now consider a user, Silva, lives in Rio de Janeiro and carries a smartphone running our Internet of Things middleware ContextNet [6], [7]. This middleware uses the

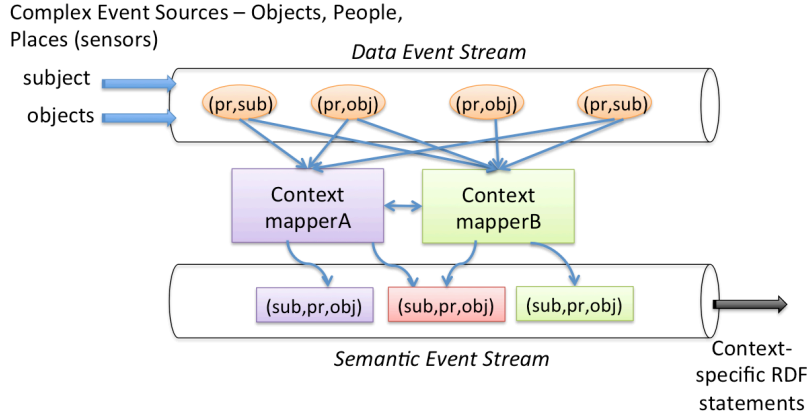


Fig. 1. Mapping Data Events to Semantic Events

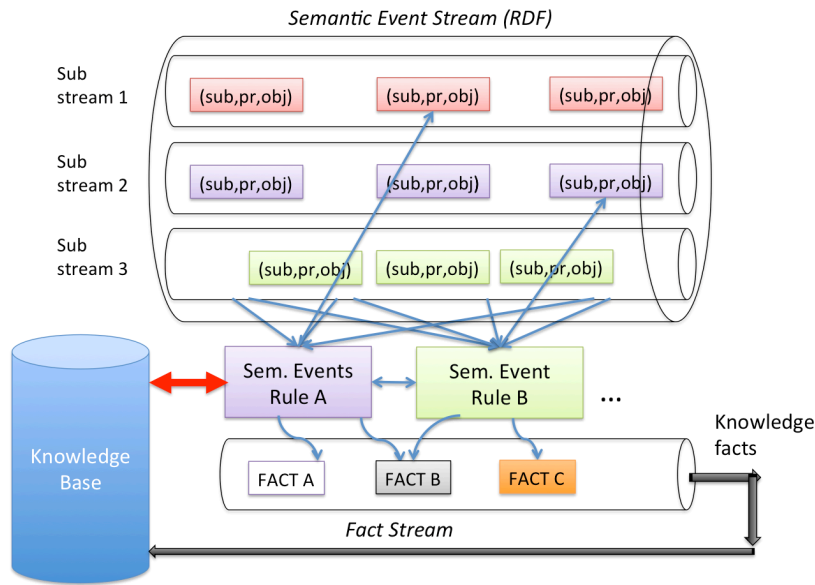


Fig. 2. Mapping Semantic Events to Knowledge Facts

smartphone as the bridge between Bluetooth-enabled smart devices/objects/sensors and IoT application servers executing in a cloud. The mobile middleware (Mobile Hub) periodically issues a BLE scan, discovers nearby BLE devices, connects to them, subscribes to the smart device’s sensors and writes commands to the smart objects that have some actuator. Figure 3 shows the Mobile Hub with four Texas Instruments *SensorTags* that have 6 different sensors, including temperature and accelerometer. Assume that it is summertime and that some IoT application needs to know if Silva is in his office, if he is walking on the street or if he is in a bus or car. Whenever Silva’s smartphone encounters a BLE smart sensor, it is possible to deduce if he is in an air conditioned space or not, and whether he is in movement or not (due to the smart device’s accelerometer). Moreover, if the location of each deployed smart device is previously registered, it is further possible to deduce if Silva is in his office or elsewhere. And

this can be deduced even without the use of GPS, either because of its signal is not available (indoors), or because Silva decides to keep it off to save the smartphone’s battery.

In this case, it would be possible to deduce the RDF triple $(Silva, rides, BusLine435)$ from the following simpler semantic events in the stream: $(Silva’s\ Portable, connectedTo, sensorX)$, $(sensorX, in, BusKKZ8674)$, $(sensorX, Abs(Accelerator) > 10)$ and $(sensorX, temp=20)$ and from the fact that $BusKKZ8674$ operates the $BusLine\ 435$. Moreover, it would be possible to deduce that Silva is moving in the traffic, but that he is in an air-conditioned bus, which may be very important during Rio’s summertime, when outdoor temperature can reach more than 45 Degrees Celsius.

V. RELATED WORK

In an early work, Adi *et al.* [8] presents abstractions that describe semantic relationships between events, object and



Fig. 3. ContextNet Mobile Hub with four SensorTags

tasks. These are defined as generalizations and associations and through attributes that may reference events. Their abstractions are suitable for specification but cannot be computed efficiently. On the other hand, the work [9] describes a system (ETALIS) that can perform reasoning over streaming events with respect to background knowledge, similar to our Knowledge Base. It implements two languages for specification of event patterns: the rule based ETALIS Language for Events (ELA), and Event Processing SPARQL. ETALIS can evaluate domain knowledge on-the-fly, thereby proving semantic relations among events and reasoning about them. Their semantic relations among events are time-based, but don't have the synchronicity requirement. Another difference is that they do not generate a RDF Stream which they check against a knowledge base. Thus, their inference is much simpler than the one proposed in our project.

Tachmazidis *et al.* [10] propose a reasoning method over RDF triples based on defeasible logic (i.e., a non-monotonic logic) which can be implemented in a massively parallel way. They used Hadoop, an open-source implementation of the MapReduce paradigm, and a stratified rule set for a more efficient processing of the knowledge base. Unlike our proposal, they do not handle Stream Processing and do not apply their method to reasoning for time-critical systems, such as CPS. Moreover, their choice for defeasible logic limits the sorts of knowledge that can be inferred by their system, as opposed to temporal logic, which shall have render highly parallelizable implementations.

The following projects CityPulse [11], Star-City [12] and FIESTA-IoT [13] also present research toward the use of Semantic Stream reasoning. All of these projects use the knowledge base in order to deduce new context/facts. Also, they use a single-layer (or flat) ontology model, which differs from our ontology model that is organized as nested contexts. Moreover, none of these projects focus on the problem of delivering real-time reasoning.

The FIESTA-IoT project [13] integrates several other projects and one of them is the CityPulse project [11]. The

main goal of these projects is to achieve semantic interoperability at different levels (hardware, data, model, query, reasoning and application levels). The StarCity project has a similar idea, but it is aimed at using semantics to provide interoperability at the data level.

On the other hand, the work by Teymourian *et al.* [14] has the same focus as our work. They use a similar idea and combine the use of SCEP rules (Semantic Web plus CEP) with a semantic knowledge base to deliver real-time reasoning. The difference is that our work uses an ontology model organized as nested context to represent context information, rather than a flat ontology model. As a result, it could be more efficient on query processing, because when we execute a query, the query will be processed only using a sub set of the knowledge base (a partition of the knowledge base). Furthermore, other difference is that we purpose to insert new SCEP rules on-the-fly, based on new facts generated by the reasoning over the knowledge base. Consequently, it is a more efficient approach for the application to adapt to different situations. For example, in a monitoring application, we only need a CEP rule that triggers an action based on an altitude situation only if the monitored person is in a high altitude, until then this rule does not need to be there.

VI. DISCUSSION

Combining symbolic reasoning based on Ontologies with Complex Event Processing has several advantages. Firstly, it allows to leverage CEP efficient processing of dense flows of simple events, not just over raw sensor events but also over RDF triples. Secondly, CEP's ability to produce complex events is also necessary for the iterative generation of higher level information from lower level bits of information.

On the other hand, while CEP is appropriate for processing data that is carried by the incoming events, it is incapable of detecting domain-specific relationships between events that are produced by distinct entities/objects that apparently have no relation with each other, or when this relationship cannot be directly encoded by the (meta-)information carried by the events. Symbolic reasoning using ontologies, on the

other hand, can very well model these “indirect” relationships among the monitored entities and/or their corresponding events. And hence, by using the results of a query over a domain-specific ontology during a CEP-based continuous processing, it becomes possible to generate new sorts of events (i.e., fact events), which are produced independently by the Semantic Event reasoners in response to the consumption of some RDF triples. These Fact events, which in some sense embody some semantic knowledge that was forked off the knowledge base, can in turn be further processed by other CEP engines and may be used to predict events that actually did not yet happen, but which are a natural consequence of initial events that have been detected by CEP.

This makes us consider the Semantic Web reasoners as a special kind of CEP engines, which have access to knowledge base, consume RDF events and eventually produce fact events that are passed onto other CEP engines in the Event Processing Network. See Figure 1 for a general architecture.

VII. CONCLUSION AND FUTURE WORK

This paper presented a real-time reasoning approach based on semantic events and fact streams for IoT systems. The reasoning approach is based on the assumptions that all objects, people, buildings, places, vehicles, environments, etc. will have many embedded tiny sensors that will emit simple events whenever some action is performed with/to it by an actor and that each event will carry the items’ unique UUID and an accurate time-stamp. By enforcing the restriction that predicates in a RDF triple must be action-based, such as `kick`, `put`, `grab`, etc., rather than state-based, such as `has`, `is`, `belongs to`, etc., we are of course limiting the amount of information that the data/event streams are capable to express. However, we believe that the action-based predicates are the really important ones for reasoning in IoT applications. All the state predicates, on the other hand, should instead be represented by the nested context-based ontology in the Knowledge Base.

We are aware that this is only a first and initial step towards adding semantics to real-time reasoning over data streams and that much more theoretical and practical research is required to validate our approach, evaluate it under a broader perspective and show its feasibility for large-scale and distributed IoT applications. However, we are confident that it is a promising first step. As next steps, we will develop simple Context Mappers and Semantic Event Rules using Esper’s EPL and deploy them on the mobile component *Mobile Hub* and on cloud-based Complex Event Processing engines in the cloud of our IoT middleware. In parallel, we will model a simple scenario and the main entities and their relationships, as described in Section IV and represent it as nested contexts.

Last, we have already started investigating various machine learning techniques (autoencoders, k -means, Markov chains, recurrent networks and inductive logic programming) to automate features extraction, clustering, temporal patterns induction and knowledge induction, as ways to automatically extract useful knowledge at various stages of our architecture. One

important issue is the dynamicity of the data produced, i.e. we need to find good trade-offs depending on the usage, the nature of the data and the computing and communicating resources available, between the demand for higher level knowledge, the cost for extracting it and the risk of it being obsolete because of the constant stream of data. This preliminary study will be presented in another paper to appear.

ACKNOWLEDGEMENT

Our ESMOCYP (Efficient Semantic Models and Fault-tolerant Middleware for CYber-Physical Systems) Project is being supported by CAPES (Brazil) and DAAD (Germany) cooperation program PROBRAL (Process No 8148/2015-05) and by a CAPES PVE fellowship to J.-P. Briot. We would like to thank the researchers and students from the University of Stuttgart for fruitful discussions.

REFERENCES

- [1] D. C. Luckham, *The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems*. Boston, MA, USA: Addison-Wesley, 2001.
- [2] K. S. Candan, H. Liu, and R. Suvarna, “Resource description framework: Metadata and its applications,” *SIGKDD Explor. Newsl.*, vol. 3, no. 1, pp. 6–19, Jul. 2001.
- [3] A. Arasu, S. Babu, and J. Widom, “The CQL continuous query language: Semantic foundations and query execution,” *The VLDB Journal*, vol. 15, no. 2, pp. 121–142, Jun. 2006.
- [4] O. Etzion and P. Niblett, *Event Processing in Action*, 1st ed. Greenwich, CT, USA: Manning Publications Co., 2010.
- [5] I. Cafezeiro, J. Viterbo, A. Rademaker, E. H. Haeusler, and M. Endler, “Specifying ubiquitous systems through the algebra of contextualized ontologies,” *The Knowledge Engineering Review*, vol. 29, no. 02, pp. 171–185, 2014.
- [6] M. Endler, G. Baptista, L. D. Silva, R. Vasconcelos, M. Malcher, V. Pantoja, V. Pinheiro, and J. Viterbo, “ContextNet: Context reasoning and sharing middleware for large-scale pervasive collaboration and social networking,” in *ACM/IFIP/USENIX 12th International Middleware Conference – Workshop, Posters and Demos Track*. Lisbon, Portugal: ACM, Dec. 2011, pp. 2:1–2.
- [7] L. E. Talavera, M. Endler, I. Vasconcelos, R. Vasconcelos, M. Cunha, and F. Silva, “The Mobile Hub concept: Enabling applications for the internet of mobile things,” in *International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*. IEEE, March 2015, pp. 123–128.
- [8] A. Adi, D. Botzera, and O. Etzion, “Semantic Event Model and its Implication on Situation Detection,” in *European Conference on Information Systems (ECIS 2000)*, 2000, Paper 2.
- [9] D. Anicic, S. Rudolph, P. Fodor, and N. Stojanovic, “Stream Reasoning and Complex Event Processing in ETALIS,” *Semantic Web*, no. 1, pp. 1–5, 2009.
- [10] I. Tachmazidis, G. Antoniou, G. Flouris, S. Kotoulas, and L. McCluskey, “Large-scale parallel stratified defeasible reasoning,” in *20th European Conference on Artificial Intelligence (ECAI’2012)*, Montpellier, France, Aug. 2012, pp. 738–743.
- [11] M. Giatsoglou, D. Chatzakou, V. Gkatziaki, A. Vakali, and L. Anthopoulos, “CityPulse: A platform prototype for smart city social data mining,” *Journal of the Knowledge Economy*, vol. 7, no. 2, pp. 344–372, 2016.
- [12] F. Lécué, S. Tallevi-Diotallevi, J. Hayes, R. Tucker, V. Bicer, M. L. Sbodio, and P. Tommasi, “Star-City: semantic traffic analytics and reasoning for city,” in *19th International Conference on Intelligent User Interfaces (IUI’14)*. Haifa, Israel: ACM, Feb. 2014, pp. 179–188.
- [13] Y. Al-Hazmi and T. Magedanz, “Towards semantic monitoring data collection and representation in federated infrastructures,” in *3rd International Conference on Future Internet of Things and Cloud (FI-Cloud’2015)*. IEEE, 2015, pp. 17–24.
- [14] K. Teymourian, M. Rohde, and A. Paschke, “Fusion of background knowledge and streams of events,” in *6th ACM International Conference on Distributed Event-Based Systems (DEBS’12)*. ACM, 2012, pp. 302–313.