

Web Stream Reasoning: From Data Streams to Actionable Knowledge

Alessandra Mileo^(✉)

Insight Centre for Data Analytics, National University of Ireland, Galway, Ireland
alessandra.mileo@insight-centre.org

Abstract. A fast growing torrent of data is being created by companies, social networks, mobile phones, smart homes, public transport vehicles, healthcare devices, and other modern infrastructures. Being able to unlock the potential hidden in this torrent of data would open unprecedented opportunities to improve our daily lives that were not possible before. Advances in the Internet of Things (IoT), Semantic Web and Linked Data research and standardization have already established formats and technologies for representing, sharing and re-using (dynamic) knowledge on the Web. However, transforming data into actionable knowledge requires to cater for (i) automatic mechanisms to discover and integrate heterogeneous data streams on the fly and extract patterns for applications to use, (ii) concepts and algorithms for context and quality-aware integration of semantic data streams, and (iii) the ability to synthesize domain-driven commonsense knowledge (and answers derived from it) with expressive inference that can capture decision analytics in a scalable way. In the first part of this lecture we will characterize the main approaches to stream processing for the Web of Data, showing how data quality and context can guide semantic integration. In the second part of this lecture we will focus on rule-based Web Stream Reasoning and illustrate how scalability and uncertainty issues can be addressed in a rule-based approach. We will discuss new challenges and opportunities in Web Stream Reasoning, briefly considering economical and societal impact in real application scenarios in a smart city context, and we will conclude by providing a brief overview of ongoing research and standardization activities in this area.

Keywords: Stream reasoning · Continuous query processing · Quality of information · Logic programming · Semantic web · Inductive logic reasoning

1 Introduction

The Semantic Web and the growing interests in linking data for sharing, re-use, and understanding has started to intersect with the domain of *Big Data* [38].

This research has been partially supported by Science Foundation Ireland (SFI) under grant No. SFI/12/RC/2289 and EU FP7 CityPulse Project under grant No.603095. <http://www.ict-citypulse.eu>.

To be successful and efficient in this joint space, we must consider the impact of the volume, variety, and velocity of data on the Web similarly to the Big Data world. The use of RDF as the common data model helped in dealing with the *variety* of information, while various software technologies – such as advanced RDF triplestores – are handling the *volume* of already available data. However, the problem of *velocity*, i.e., frequently produced and *streamed* data still presents some open challenges [10, 11].

Applications that can process streaming data incrementally are required for sensor networks and the Internet of Things (IoT), Smart Grids, Smart Cities, health care and assisted living, security, social network analysis, financial planning, etc. In these domains it is not only necessary to *make sense* of the data very quickly but also to do so in the context of “static” background knowledge such as planning goals, plans, capacities and physical layouts. These real-world requirements necessitate to move the processing paradigms for vast amounts of data from the current batch-like approaches (e.g., distributed and parallel computing with MapReduce) towards processing of data streams and stream reasoning in near-real-time.

Advances on Semantic Web & Linked Data research and standards have already provided formats and technologies for representing and sharing knowledge on the Web. In the last few years, Semantic Web technologies such as RDF, OWL, SPARQL have provided mechanisms and related engines for continuously querying semantic data streams [5, 7, 22] and for semantic complex event processing [2, 20, 21]. Despite their potentials for dealing with data that changes in high volume at high frequency, these solutions can not properly deal with the noisy and imprecise nature of data in dynamic domains such as those mentioned earlier in this section, which are characterized by incomplete information, uncertainty, inconsistencies, preferences and qualitative optimization.

Dealing with these characteristics of dynamic information requires complex reasoning capabilities such as the ability of managing defaults, common-sense, preferences, recursion, and non-determinism which might be required for more expressive reasoning tasks. Logic-based non-monotonic reasoners can perform such computationally intensive tasks but available solutions are suitable for data that changes in low volumes at low frequency and therefore their applicability is limited.

This lecture will characterize IoT Intelligence solutions based on their scalability and expressivity, and will explore their synergies and potentials to be used as a pipeline for scalable and expressive Web Stream Reasoning. Approaches and techniques to handle uncertainty and context-driven information integration will also be presented.

The remainder of the material is structured as follows: Sect. 2 identifies the IoT Intelligence layers considering their expressivity and scalability based on the underlying semantics of existing systems. Section 3 provides some pointers and principles for RDF stream processing and Semantic Complex Event Processing, touching upon quality-aware information integration. Section 4 focuses on the non-monotonic reasoning layer and discusses the latest directions in this area,

including hybrid mechanisms where non-monotonic logics and inductive reasoning are combined to deal with uncertainty. Section 5 concludes by presenting recent developments on formal generalizations and standards.

2 IoT Intelligence Layers

Scenarios and requirements for Stream Reasoning have been presented in [27] considering applications for smart grids and smart cities, health monitoring, social media and logistics among others. If we consider existing approaches and solutions for transforming IoT data produced as web streams into knowledge, we can characterize them into three main layers based on the expressivity of the reasoning tasks they support. The conceptual representation of these layers is indicated in Fig. 1. Several interesting approaches are flourishing, which try to extend existing systems for web stream reasoning with cross-layer features. However, we argue existing solutions can be associated to one of these conceptual layer:

Stream Query Processing Layer: This layer includes systems which rely on SPARQL extensions to deal with streaming data. In principle they support all the features and operators of SPARQL 1.1, although implementations might vary, and they have the ability to process and semantically integrate static and dynamic Linked Data.

Example 1. Let us consider data about taxis in a smart city (inspired by the last DEBS Grand Challenge¹ based on NYC open data). Finding the most frequent routes, the most profitable pick-up or drop-off point, the neighborhoods in which pick-up/drop-off increased, or comparing taxi rides with areas served with public transportation are all examples of stream query processing, where dynamic data streams about taxi rides, and static linked data about bus routes or GeoNames need to be semantically integrated.

Semantic Complex Event Processing (SCEP) Layer: Systems in this layer aim at combining stream query processing with operators for complex event pattern detection. These approaches are mostly based on rules for pattern detection using logical operators, and go beyond the current support provided by stream query processing engines to the SPARQL 1.1 semantics. Approaches and systems for semantic complex event processing have leveraged engines for stream query processing and complex event processing in combination, in order to achieve better trade-offs when it comes to expressivity vs. scalability.

Example 2. Let us consider a Social Sensing scenario where we aim at detecting some specific patterns in the interactions among people. In order to detect the most active subjects (e.g. subjects that have been in more than 10 interactions in the last half an hour) stream query processing with aggregates would be enough. But if we want to detect whenever two subjects have moved from one

¹ <http://www.debs2015.org/call-grand-challenge.html>.

room X to another room Y , maybe counting how many times this happened for two specific rooms or for two specific subjects, or finding all the sequences of rooms $\langle X, Y \rangle$ for which the counting is higher than a threshold, then we fall into complex event processing and need to make sure certain operators are supported: we need to keep track of the status of certain events (i.e. a subject being with another subject and moving from a room to another) and identify sequences and repetitions of such events.

Stream Reasoning Layer: This layer is concerned with approaches to producing new logical conclusions from a given set of input facts, by applying a set of rules. It is the more expressive and less explored layer of web stream reasoning, and it includes approaches that are able to deal with uncertainty, non-monotonicity, defaults and common sense inference. In this lecture we consider rule-based approaches to non-monotonic stream reasoning and presents some principles and directions in this area.

Example 3. Let us consider a geo-fencing scenario similar to the one described in [28]. People wear RFID tags and move around a building or an area such as an airport or a shopping mall, equipped with RFID readers producing streams of position information. Within the area, we have defined “geo-fences”, i.e., virtual perimeter for a real-world area which are used to mark particular spaces as “off-limits”. Rule-based inference that considers conflicts, non-monotonicity, and uncertainty are required to detect when a particular area is at risk and what are the different ways somebody could breach the geo-fence. When we introduce noise in the sensory input, and constraints based on adjacency of certain areas, conflicts can be detected and noise needs to be filtered out. This can be done with rule-based approach by encoding optimization (e.g. minimizing the error) or by using probabilistic approaches to rule-based inference.

Cross-Layer Processing: Recent approaches attempt to improve scalability by relying on systems from the underlying layers to filter and aggregate sensor data into events or complex events, and then use results of this pre-processing to perform complex inference. For example, few approaches have combined SCEP systems with production rules systems [32, 36], although they often trade expressiveness for response time. Relying on underlying mechanisms for Stream Query Processing in order to filter relevant data has also been considered as a way to reduce the size of the input for the more expressive layers, as in the combination with Stream Query Processing and Answer Set Programming [28].

In this lecture we are mainly concerned with the following requirements from real world applications:

- **Expressivity:** Deduction processes aim at deriving knowledge from data, and the underlying semantics dictates how complex and expressive an inference language is; application scenarios that require to deal with default knowledge, preferential and probabilistic rules, non-determinisms and recursion require more expressive stream reasoning formalisms that are sitting at the top layer of IoT Intelligence, identified as Layer 3 in Fig. 1;

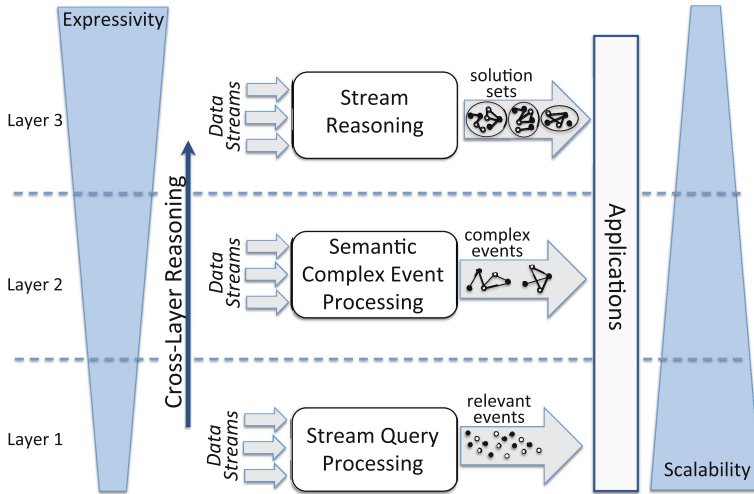


Fig. 1. IoT intelligence layers

- **Efficiency:** Some real world applications demand for low latency processing and require a timely response; this can be challenging with high volumes of incoming data, since it requires to design solutions that can achieve low latency and high throughput, possibly sacrificing expressivity;
- **Quality-Aware Stream Processing:** When it comes to application and services, quality constraints and requirements might vary; being able to identify the quality of a stream, being it part of input data or resulting from a processing step, Quality of Information (QoI) can play a crucial role not only in providing better solutions but also in solving inconsistencies and conflicts;
- **Uncertainty Management:** IoT data can be incomplete, contradictory and noisy, which requires to deal with uncertainty and approximation without losing structural and causal connections between data and event streams.

These requirements have been only partially addressed in existing systems across the three layers. As part of this lecture, we will provide an overview of to what extent existing approaches to IoT Intelligence meet these requirements, and this will help identifying the gaps in existing solutions for Web Stream Reasoning².

3 RDF Stream Processing

The ability to process RDF streams requires to adapt the RDF data model to capture data items that flow continuously over time, forming unbounded sequences of data. To date several stream processing engines have been proposed for processing RDF streams as Linked Data and the semantic web community

² Slides will be available for download from <http://www.streamreasoning.org/events/>.

has been active in this area, defining vocabularies and languages to represent and process RDF streams.

As a consequence, more and more semantic data streams have appeared on the open, loosely governed and heterogeneous Web environment, increasing dramatically the potentials for observable events to be captured and processed. This attracted the attention of the CEP community and the Semantic Web community to join forces towards bridging this semantic gap.

Advances in Semantic Web and Linked Data research and standardization have established formats and technologies for representing, sharing and re-using knowledge on the Web, including streaming data such as social content and the Internet of Things [33]. As a result, the Web of Data is today overwhelmed with events, which has contributed to an unprecedented shift in the quantity and quality of dynamic information enabling complex knowledge to be linked and available for processing.

Acknowledging the need of semantics for better interpretation of such a massive amount of events, the Semantic Web community has moved towards Semantic Complex Event Processing (or SCEP) which uses ontological models to filter, aggregate and interpret complex events based on their semantic correlation. Beyond the continuous identification of complex semantic events via query processing, the need for more expressive rules to enhance reasoning capabilities in transforming events to actionable knowledge has also been recently investigated, as well as the introduction of mechanisms to deal with noisy data by using quality-aware complex event processing.

In the remainder of this section we will provide a quick overview and a few pointers on RDF stream processing and quality-aware event composition.

3.1 Linked Streams Data Processing

As Linked Data facilitates the data integration process among heterogeneous collections, Linked Stream Data has the same goal with respect to data streams. Considering streams as another form of Linked Data bridges the gap between dynamic and static data sources, and makes it possible to query and integrate them in a single framework.

Stream query processing is under active research for several years in Database as well as in the Semantic Web community [5, 8, 22, 25] and interesting solutions have been proposed to process static and dynamic structured data via continuous queries [5–7, 22].

Unlike query processing for linked datasets which is mostly pull-based and one-time only, in Linked Stream Data processing new data items are produced continuously, the data is often valid only during a time window, and it is continually pushed to the query processor. Queries are continuous, i.e., they are registered once and then are evaluated continuously over time against the changing dataset. The results of a continuous query are updated as new data appears. We refer the reader to [23] for an overview of Linked Stream Data processing, which highlights basic requirements, language syntax and semantics, different processing methods and the advantages and disadvantages of existing approaches.

3.2 Semantic Complex Event Processing (SCEP)

The combination of Complex Event Processing (CEP) and semantic technologies plays a key role in enabling IoT Intelligence in such a way to improve flexibility and expressivity of current Linked Stream Data processing. There is a need to cater for available background knowledge when detecting and responding to complex events, motivated in many application scenarios where it is important to seamlessly integrate changes into CEP systems, translating events, patterns and reactions into operations in a declarative way.

Semantic Complex Event Processing (SCEP) [9,34] started in recent years, and a number of systems exist [1,2,20,21]. These systems support operators that are not natively implemented in Linked Stream Data processing engines, such as the ability to detect complex event patterns as sequences, temporally ordered events and repetitions. Unlike stream query processing systems, SCEP engines do not have the ability to process structured streams as Linked Data, but they support background knowledge and some form of (monotonic) reasoning.

For these reasons, in the scope of this lecture we position them in a different layer and we separate them from non-monotonic reasoning approaches, which we will be investigating more in details in Sect. 4.

Rule-based SCEP has been investigated in the last decade, with a growing scientific community that is also active in standardization activities. This includes initiatives around RuleML and reaction rules [32] as well as Prolog-based approaches for processing complex events [31]. We invite the readers to consult surveys and tutorials on SCEP available at http://wiki.ruleml.org/index.php/Reaction_RuleML.

3.3 Quality-Aware SCEP

SCEP has been proved to be efficient for processing streams with high frequency and complex query semantics. Recent developments in Internet-of-Things (IoT) and Smart City applications bring new challenges to conventional SCEP systems, e.g., incorporating heterogeneous event sources, formats or event stream processing engines. Moreover, there is a need to explore automatic ways to recover the system from erroneous states, and to discover and compose event streams according to application requirements and constraints. Solving this problem often boils down to automatically discover what streaming sources can best answer complex event requests and identify which event source should be considered to match specific quality requirements from users and applications.

Non-functional properties, e.g.: quality-of-service (QoS) properties, can play a pivotal role in guiding such selection if used as dimensions for finding the optimal event service composition plan that provides the best available results. Existing publish/subscribe based event systems and middleware use proprietary event advertisement and subscription formats (which leads to silo architectures) and provide limited supports for non-functional requirements related to event subscriptions [26].

To address these issues, a body of work has been proposed that integrates SCEP systems with Service Oriented Architecture (SOA) [15]. This approach directly addresses the problem of dealing with data quality of streams and uses it not only to provide the best available semantic complex event plan, but also to support the engineering side of practical deployments by helping to plan what performance parameters work best under a given input load.

4 Web Stream Reasoning

Stream Reasoning for the (Semantic) Web is mainly concerned with the ability to deal with the imperfect nature of web streams, so that inference algorithms can be successfully applied to a variety of real-world applications. As mentioned in Sect. 1, streaming sources can sometimes behave erratically and generate incomplete and noisy data. Without proper mechanisms, stream reasoning systems can then be caught up in attempting to deal with situations involving conflicting knowledge (e.g. temperature sensors providing a value of 20C and fire detectors alerting of a fire). Even worse, a system can end up failing when it enters an undecidable reasoning state due to contradiction or non-determinism. This happens when there are several possible conclusions or solutions as a result of given observations, or when there is no outcome satisfying all given constraints. For example each traffic light in a crossing can be red, yellow or green in different combinations, and there are constraints on synchronization between them; similarly, there are different possible paths for going from A to B and there might be constraints and preferences on time, distance, CO2 intake, safety of the road etc. that determine which solution is best. Non-monotonic formalisms can help dealing with logical contradiction, incompleteness and non-determinisms in stream reasoning by embracing incomplete and noisy streams and presenting results as a set of plausible (possibly ranked) solutions. This leads to a system which is more robust and expressive than any current stream reasoning implementation for the (Semantic) Web. As a result, Non-Monotonic Reasoning (NMR) techniques for (Semantic) Web Streams can be seen as having high potential impact in a variety of real-world applications.

The ability of dealing with incomplete and noisy input streams is one of the capabilities induced by non-monotonicity, but providing support for dealing with conflicts, defaults, qualitative preferences, constraints, and non-determinism requires computationally intensive reasoning.

A few approaches have been investigated that aims at supporting NMR for big data. The prominent categories of such approaches rely on either the Well-Founded Semantics (WFS) and defeasible reasoning, or the Stable Model Semantics and Answer Set Programming (ASP). Given the complexity of NMR reasoning over streams, cross-layer approaches that leverage processing at different level of complexity is recently being investigated. In what follows we briefly summarize the approaches in each of these categories, that will be covered in this lecture.

4.1 Large-Scale Defeasible Reasoning with MapReduce

Authors in [3,35] focus on distributed methods for non-monotonic rule-based reasoning. Their current works perform parallel defeasible reasoning under the assumption of stratification which imposed a severe limitation considering the range of allowed rule set. Also, they focus on optimization of WFS computation based on MapReduce. Despite these approaches might have computational advantages over the more complex ASP-based approaches, the implementation based on MapReduce makes them suitable for embarrassingly parallel problems but not for problems with exponential complexity. Additionally, the available implementations based on MapReduce do not natively support stream processing concepts such as time-decay model and sliding window, making it less intuitive to specify problems in terms of stream reasoning tasks. We will briefly illustrate the core idea behind these approaches.

4.2 Web Stream Reasoning with Answer Set Programming

Developments on the Datalog side are evolving in this directions, and extensions of Datalog towards the logic paradigm of Answer Set Programming (ASP) [4,17,24] have been implementing these reasoning capabilities which can go far beyond the capabilities of existing query engines. Logic programming dialects like Datalog with negation, covered by ASP, are viewed as a natural basis for the Semantic Web rule layer [13], but the full expressivity of ASP introduces new challenges concerning the trade-off between expressivity and scalability, especially in a streaming scenario. Therefore, when dealing with NMR approaches based on ASP, particular attention should be given to the scalability of such systems. The development of stream reasoning systems based on the Stable Model Semantics focuses on extending the well established declarative complex reasoning framework of ASP with dynamic data. M. Gebser et al. [16] proposed modeling approaches for continuous stream reasoning based on reactive ASP, utilizing time-decaying logic programs to capture sliding window data in a natural way. This is a first step towards gearing ASP to continuous reasoning tasks. However, these approaches still mainly process on low changing data and relatively smaller data sizes. Do et al. [12] also utilize ASP in their stream reasoning system and the approach is based on the DLV engine [14], which does not deal with continuous and window-based reasoning over data stream within the reasoner.

4.3 Cross-Layer Web Stream Reasoning with ASP

NMR for Semantic Web Streams has only started to be investigated in recent years and no commercial systems beyond a few small-scale research prototypes exist. There is little scientific work which tries to capitalize on the synergies between stream query processing and stream reasoning and there is a quickly growing demand for software solutions that can efficiently process web streams and perform complex reasoning tasks on noisy and incomplete input. A similar

approach is proposed in [28], where the authors present the StreamRule framework as a combination of linked stream data processing and NMR in ASP.

In this lecture we will mostly focus on ASP-based approaches to NMR, relying on existing solvers that support stream processing features and uncertainty management via rule learning. As mentioned earlier in this section, ASP-based approaches are computationally more expensive than parallel approaches based on defeasible reasoning, but they are suitable for problems with exponential complexity. We will investigate a new line of research that leverages cross-layer processing of streams, combining approaches across the three layers of Fig. 1. Our main assumption is that we can efficiently perform NMR by utilizing approaches from both stream processing and stream reasoning, when combined correctly under a common and sound model. Focusing on NMR methods, we will explore approaches and open challenges for web stream reasoning which rely on the synergies between RDF stream processing and rule-based inference. The two main directions we will consider in this lecture are:

- Combined approaches that rely on web stream reasoning layers at lower complexity to reduce the size of the input and increase scalability at the higher levels [18, 28];
- Hybrid approaches to uncertainty management, which combine declarative non-monotonic reasoning with inductive inference and learning [29, 30].

We will provide an overview of prototypical tools and showcase how they can be used in a smart city context³.

5 Conclusive Remarks

In this lecture we provide an overview of Web Stream Reasoning, considered as the application of reasoning techniques to help deriving actionable knowledge from web data streams. Stream reasoning is an unexplored yet high impact research area and encompasses a series of new multidisciplinary approach that can provide the abstractions, foundations, methods, and tools required to integrate data streams, semantic representations, complex events, and reasoning systems [37].

A variety of concrete applications highlight clearly the need for scalable web stream reasoning and the importance of characterizing the expressivity vs. scalability trade-off to tackle the efficiency and expressivity challenges. Approaches that incrementally filter, process and aggregate web streams to enable higher level inference are in their infancy and they are only one possible direction to address such challenges. Even though IoT intelligence in modern applications often requires expressive and scalable languages and methods for web stream reasoning, current approaches rely on different underlying formalisms which require the use of an external reasoner and expensive mapping and synchronization between the different layers, with consequent negative impact on scalability.

³ <http://www.ict-citypulse.eu>.

Promising research activities are ongoing to address these challenges. Some of them worth mentioning include the DHSR project⁴ and the W3C RDF Stream Processing Working Group (RSP WG)⁵. The DHSR project aims at providing a strong model-based semantic foundation to distributed heterogeneous stream reasoning. RSP WG standardization activities are fostering the semantic community to define a common and extensible core model for RDF stream processing, envisioning an ecosystem of streaming and static RDF data sources whose data can be combined through standard models, languages and protocols. Relevant research is being carried forward in the context of the EU FP7 project CityPulse, where mechanisms for adaptive RDF stream processing and dynamic data-driven heuristics for scalable NMR over streams are being investigated [19].

References

1. Anicic, D., Fodor, P., Rudolph, S., Stojanovic, N.: ET-SPARQL: a unified language for event processing and stream reasoning. In: Proceedings of the 20th WWW Conference, pp. 635–644, ACM (2011)
2. Anicic, D., Rudolph, S., Fodor, P., Stojanovic, N.: Stream reasoning and complex event processing in etalis. *Semant. Web* **3**(4), 397–407 (2011)
3. Antoniou, G., Batsakis, S., Tachmazidis, I.: Large-scale reasoning with (semantic) data. In: Proceedings of the 4th International Conference on Web Intelligence, Mining and Semantics (WIMS 2014), p. 1, ACM (2014)
4. Baral, C.: Knowledge Representation Reasoning and Declarative Problem Solving. Cambridge University Press, Cambridge (2003)
5. Barbieri, D.F., Braga, D., Ceri, S., Valle, E.D., Grossniklaus, M.: Querying rdf streams with C-SPARQL. *SIGMOD Rec.* **39**(1), 20–26 (2010)
6. Bolles, Andre, Grawunder, Marco, Jacobi, Jonas: Streaming SPARQL - Extending SPARQL to process data streams. In: Bechhofer, Sean, Hauswirth, Manfred, Hoffmann, Jörg, Koubarakis, Manolis (eds.) *ESWC 2008. LNCS*, vol. 5021, pp. 448–462. Springer, Heidelberg (2008)
7. Calbimonte, J., Jeung, H., Corcho, Ó., Aberer, K.: Enabling query technologies for the semantic sensor web. *Int. J. Semant. Web Inf. Syst.* **8**(1), 43–63 (2012)
8. Carney, D., Çetintemel, U., Cherniack, M., Convey, C., Lee, S., Seidman, G., Stonebraker, M., Tatbul, N., Zdonik, S.: Monitoring streams: a new class of data management applications. In: *VLDB 2002*, pp. 215–226, VLDB Endowment (2002)
9. Della Valle, E., Ceri, S., Barbieri, D.F., Braga, D., Campi, A.: A First Step Towards Stream Reasoning. In: Domingue, J., Fensel, D., Traverso, P. (eds.) *FIS 2008. LNCS*, vol. 5468, pp. 72–81. Springer, Heidelberg (2009)
10. Della Valle, E., Ceri, S., van Harmelen, F., Fensel, D.: It’s a streaming world! reasoning upon rapidly changing information. *IEEE Intell. Syst.* **24**(6), 83–89 (2009)
11. Della Valle, E., Schlobach, S., Krötzsch, M., Bozzon, A., Ceri, S., Horrocks, I.: Order matters! harnessing a world of orderings for reasoning over massive data. *J. Semant. Web* **4**(2), 219–231 (2012)
12. Do, Thang M., Loke, Seng W., Liu, Fei: Answer set programming for stream reasoning. In: Butz, Cory, Lingras, Pawan (eds.) *Canadian AI 2011. LNCS*, vol. 6657, pp. 104–109. Springer, Heidelberg (2011)

⁴ <http://www.kr.tuwien.ac.at/research/projects/dhsr/>.

⁵ <http://www.w3.org/community/rsp/>.

13. Eiter, T., Ianni, G., Polleres, A., Schindlauer, R., Tompits, H.: Reasoning with rules and ontologies. In: Barahona, P., Bry, F., Franconi, E., Henze, N., Sattler, U. (eds.) Reasoning Web 2006. LNCS, vol. 4126, pp. 93–127. Springer, Heidelberg (2006)
14. Eiter, T., Ianni, G., Schindlauer, R., Tompits, H.: Dlv-hex: Dealing with semantic web under answer-set programming. In: The Proceedings of the 4th International Semantic Web Conference (2005)
15. Gao, F., Curry, E., Ali, M.I., Bhiri, S., Mileo, A.: QoS-Aware complex event service composition and optimization using genetic algorithms. In: Franch, X., Ghose, A.K., Lewis, G.A., Bhiri, S. (eds.) ICSSOC 2014. LNCS, vol. 8831, pp. 386–393. Springer, Heidelberg (2014)
16. Gebser, M., Grote, T., Kaminski, R., Obermeier, P., Sabuncu, O., Schaub, T.: Answer set programming for stream reasoning (2013). CoRR [abs/1301.1392](https://arxiv.org/abs/1301.1392)
17. Gelfond, M., Lifschitz, V.: The stable model semantics for logic programming. In: Proceedings of the 5th International Conference on Logic Programming, vol. 161 (1988)
18. Germano, S., Pham, T.-L., Mileo, A.: Web stream reasoning in practice: on the expressivity vs. scalability tradeoff. In: Web Reasoning and Rule Systems - 9th International Conference, RR 2014, Berlin, Germany, 5–6 August 2015, page to appear. Proceedings (2015)
19. W. S. R. in Practice: on the Expressivity vs. Scalability tradeoff. Stefano germano and thu-le pham and alessandra mkileo. In: Web Reasoning and Rule Systems - 9th International Conference, RR 2015, Berlin, Germany, 4–5 August 2015, page to appear. Proceedings (2015)
20. Komazec, S., Cerri, D., Fensel, D.: Sparkwave: continuous schema-enhanced pattern matching over rdf data streams. In: Proceedings of the 6th ACM International Conference on Distributed Event-Based Systems, pp.58–68, ACM (2012)
21. Lanzasasto, N., Komazec, S., Toma, I.: Reasoning over real time data streams (2012). http://www.envision-project.eu/wp-content/uploads/2012/11/D4-8_v1-0.pdf
22. Le-Phuoc, D., Dao-Tran, M., Xavier Parreira, J., Hauswirth, M.: A native and adaptive approach for unified processing of linked streams and linked data. In: Aroyo, L., Welty, C., Alani, H., Taylor, J., Bernstein, A., Kagal, L., Noy, N., Blomqvist, E. (eds.) ISWC 2011, Part I. LNCS, vol. 7031, pp. 370–388. Springer, Heidelberg (2011)
23. Le-Phuoc, D., Xavier Parreira, J., Hauswirth, M.: Linked stream data processing. In: Eiter, T., Krennwallner, T. (eds.) Reasoning Web 2012. LNCS, vol. 7487, pp. 245–289. Springer, Heidelberg (2012)
24. Lifschitz, V.: Answer set programming and plan generation. *AI* **138**(1), 39–54 (2002)
25. Madden, S., Shah, M., Hellerstein, J.M., Raman, V.: Continuously adaptive continuous queries over streams. In: 2002 ACM SIGMOD International Conference on Management of Data, pp. 49–60, ACM, New York (2002)
26. Mahambre, S.P., Kumar, M., Bellur, U.: A taxonomy of qos-aware, adaptive event-dissemination middleware. *IEEE Internet Comput.* **11**(4), 35–44 (2007)
27. Margara, A., Urbani, J., van Harmelen, F., Bal, H.: Streaming the web: Reasoning over dynamic data. *Web Semant.: Sci. Serv. Agents World Wide Web* **25**, 24–44 (2014)
28. Mileo, A., Abdelrahman, A., Policarpio, S., Hauswirth, M.: StreamRule: A non-monotonic stream reasoning system for the semantic web. In: Faber, W., Lembo, D. (eds.) RR 2013. LNCS, vol. 7994, pp. 247–252. Springer, Heidelberg (2013)

29. Nickles, M., Mileo, A.: Probabilistic inductive logic programming based on answer set programming (2014). CoRR [abs/1405.0720](https://arxiv.org/abs/1405.0720)
30. Nickles, M., Mileo, A.: Web stream reasoning using probabilistic answer set programming. In: Kontchakov, R., Mugnier, M.-L. (eds.) RR 2014. LNCS, vol. 8741, pp. 197–205. Springer, Heidelberg (2014)
31. Paschke, A.: Rules and logic programming for the web. In: Polleres, A., d’Amato, C., Arenas, M., Handschuh, S., Kroner, P., Ossowski, S., Patel-Schneider, P. (eds.) Reasoning Web 2011. LNCS, vol. 6848, pp. 326–381. Springer, Heidelberg (2011)
32. Paschke, A., Boley, H.: Rule responder: Rule-based agents for the semant. pragmatic web. *Int. J. Artif. Intell. Tools* **20**(6), 1043–1081 (2011)
33. Sheth, A., Henson, C., Sahoo, S.S.: Semantic sensor web. *IEEE Internet Comput.* **12**(4), 78–83 (2008)
34. Stuckenschmidt, H., Ceri, S., Della Valle, E., Van Harmelen, F., di Milano, P.: Towards expressive stream reasoning. In: Proceedings of the Dagstuhl Seminar on Semantic Aspects of Sensor Networks (2010)
35. Tachmazidis, I., Antoniou, G., Faber, W.: Efficient computation of the well-founded semantics over big data (2014). CoRR [abs/1405.2590](https://arxiv.org/abs/1405.2590)
36. Teymourian, K., Rohde, M., Paschke, A.: Fusion of background knowledge and streams of events. In: Proceedings of the 6th ACM International Conference on Distributed Event-Based Systems, DEBS 2012, pp. 302–313. ACM, New York (2012)
37. Valle, E.D., Ceri, S., Harmelen, F.V., Fensel, D.: It’s a streaming world! reasoning upon rapidly changing information. *IEEE Intell. Syst.* **24**(6), 83–89 (2009)
38. Zaino, J.: Big data and the semantic web: Their paths will cross. http://semanticweb.com/big-data-and-the-semantic-web-their-paths-will-cross_b32027