Stream Reasoning Agents

Blue Sky Ideas Track

Riccardo Tommasini
Politecnico di Milano
Milano, Italy
riccardo.tommasini@polimi.it

Davide Calvaresi
HES-SO Valais-Wallis
Sierre, Switzerland
davide.calvaresi@hevs.ch

Jean-Paul Calbimonte
HES-SO Valais-Wallis
Sierre, Switzerland
jean-paul.calbimonte@hevs.ch

ABSTRACT

Data streams are increasingly needed for different types of applications and domains, where dynamicity and data velocity are of foremost importance. In this context, research challenges raise regarding the generation, publication, processing, and discovery of these streams, especially in distributed, heterogeneous and collaborative environments such as the Web. Stream reasoning has addressed some of these challenges in the last decade, presenting a novel data processing paradigm that lays at the intersection among semantic data modeling, stream processing, and inference techniques. However, stream reasoning works have focused almost exclusively on architectures and approaches that assume an isolated processing environment. Therefore, they lack, in general, the means for discovering, collaborating, negotiating, sharing, or validating data streams on a highly heterogeneous ecosystem as the Web. Agents and multi-agent systems research has long developed principles and foundations for enabling some of these features, although usually under assumptions that require to be revised in order to comply with the characteristics of data streams. This paper presents a vision for a Web of stream reasoning agents, capable of sharing not only streaming data, but also processing duties, using collaboration and negotiation protocols, while relying on common vocabularies and protocols that take into account the high dynamicity of their knowledge, goals, and behavioral patterns.

CCS CONCEPTS

• **Information systems** → **Data streams**; • **Theory of computation** → **Semantics and reasoning**; • **Computing methodologies** → **Multi-agent systems**;

KEYWORDS

MAS, stream reasoning, Web streams

ACM Reference Format:

Riccardo Tommasini, Davide Calvaresi, and Jean-Paul Calbimonte. 2019. Stream Reasoning Agents. In *Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), Montreal, Canada, May 13–17, 2019,* IFAAMAS, 5 pages.

Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), N. Agmon, M. E. Taylor, E. Elkind, M. Veloso (eds.), May 13–17, 2019, Montreal, Canada.

1 INTRODUCTION

The volume and velocity at which data are produced are far beyond our limits to consume it. More and more often, data-intensive applications process and consume information continuously as it is produced, i.e., as streams of dynamic data on-the-fly. This is actually a consequence of the nature of data, the ever-growing processing requirements, and the acknowledgment that we live in a "streaming world" [14].

This revolutionary paradigm-shift is also present in the World Wide Web. From social media to the Internet of Things (IoT), rapidly changing information flows are everywhere. Web streams are not only vast, but also heterogeneous, noisy, and incomplete. Enabling query answering and reasoning over these data streams is a considerable challenge, which was initially addressed by the Web research community, as it was developed on the seminal works that constitute the field of *stream reasoning* [2, 4, 5, 23]. Stream reasoning research initially focused on the following question: *can we make sense, in real-time, of heterogeneous, vast, noisy, incomplete data stream generated in complex domains?*[14]. This question results to be inevitably vast and, thus, none of the state-of-the-art solutions claims to have it solved completely. A multitude of solutions were proposed, each tackling a specific yet relevant problem towards a renovated stream reasoning vision [17]. These include: continuous data querying using RDF streams [23], semantic complex event processing [2], incremental maintenance of materialization [32], or online inductive analysis [4]. These approaches partially solve the big picture framed by the original stream reasoning concept. However, attempts to provide a unifying model for some of them [7, 15] resulted impractical [38].

In the original Semantic Web vision, agents had a central role [8]: intelligent agents were expected to explore and filter the Web in the wild (on behalf of users) to provide high-quality content that solved complex information needs. With time, the Semantic Web vision drifted towards an open data paradigm, supported by a set of principles and standards [9]. Recent efforts push towards the direction of a more decentralized Web [34], but they still lack a more precise definition of the role of supporting Web agents. The reasons behind this unclear role of agents within the Web ecosystem are related to how the Web evolved in the last 15 years. Indeed, at the time of Berners-Lee's vision, the Web had not yet developed an API economy nor a Data driven economy.

In this paper we present the concept of *stream reasoning agents*, highly specialized intelligent computing entities that continuously cooperate on the Web to offer optimal yet explainable answers to our complex information needs. We advocate for the development of these collaborative intelligent units with stream reasoning capabilities, which would be able to observe and manipulate the Web environment reactively on our behalf. This vision highlights the importance and the need for such decentralized environment, especially within streaming data-driven scenarios, such as those related to IoT and highly dynamic information on the Web.

In the remainder of this paper we present the challenges that need to be addressed to materialize this vision (Section 2), and the different opportunities that arise by combining stream reasoning and multi-agent systems (Section 3). Then we identify some of the existing building blocks that can contribute to realizing this idea (Section 4), before presenting a research road-map for the immediate future (Section 6).

2 CHALLENGES

The vision of *stream reasoning agents* requires addressing a number of scientific and technological challenges. Although there have been research efforts related to some of these challenges [10, 24], it is still needed to study the implications of combining stream reasoning and the multi-agent paradigm.In the following, we detail these challenges and their importance for the fulfillment of our vision.

Data streams discovery and reuse. Stream reasoning agents assume that data streams flow among a set of decentralized intelligent entities. The Web is a natural environment for the exchange of streaming data, but first, it is necessary that agents discover and learn about their existence. The aim is to make sure that these streams are findable, accessible, interoperable, and reusable (FAIR [40]). Examples exist for generic dataset search, e.g. using models as DCAT [26] or Schema.org in research ecosystems like OpenAIRE, or Google Datasets. In domain specific contexts as sensor networks [13] and IoT [21, 36], ontologies have been developed to allow this type of discovery. Moreover, stream-specific vocabularies like Vocals [37] have been proposed, although their adoption is yet to be assessed. In any case, most of these previous efforts stop at dataset (or stream) discovery. In the multi-agent paradigm, the question expands to also choosing which data streams to use, or negotiate their reuse based on goals and optimization of resources. An agent can, for instance, decide on collecting metadata from different sources, and pick only some of them based on: the quality of the data, frequency, etc., or they may consider alternative sources if there is need to replace one, or to accommodate to scalability/load requirements.

Data streams publication. The challenge of data streams publication on the Web is related to similar technical and conceptual issues as those related to stream discovery. Agent systems may need to incorporate standard models and ontologies in order to be able to publish streams, but furthermore, they will need to provide the technical features to effectively make these streams available to client agents. In particular, streams have different consumption modalities, i.e. push or pull-based, and may be accessed through different protocols. The Web provides concrete options, which have evolved to what is known today as the Web of Things. HTTP SSE, WebSocket, MQTT and other technologies are examples of these options, which would need to be incorporated into stream reasoning agents. Moreover, the *agentification* of data stream publishing has the potential to further improve current streaming ecosystems. This includes the possibility of letting agents self-organize [33] to publish aggregated streaming data, based on streams produced by other agents. This can be done for different purposes, e.g. k-anonymity data protection, summarization, massive data collection, and crowd-sourcing streaming data.

Stream reasoners cooperation. Stream processing engines have evolved in the last decade, reaching an impressive level of sophistication, as well as specialization depending on the type of data and the processing goals. In this context, it remains challenging to combine different data stream engines in a coherent manner, so that their combined processing power can deliver the desired results. Cooperation among engines can be motivated by different reasons, e.g. they may require co-processing in order to cope with very large amounts of data, or to be able to scale in case of high-velocity streams. Another reason can be linked to federation requirements, when data has to be processed locally for privacy, institutional or legal constraints, or for optimization purposes. While in the past stream reasoners have tended to be federated by central governing entities, it is required to establish self-organizing mechanisms, while considering that streams flow continuously and cannot wait for synchronization and redeployment delays. Other examples of cooperation include orchestration among reasoners of different capabilities: e.g. usage of complex event processing combined with incremental reasoning.

Reasoning & negotiation. Negotiation among stream reasoners has not been studied yet, even if it may enable the establishment of flexible and efficient processing workflows. Reasoners may negotiate on entailment regimes, levels of expressiveness, report policies, etc., depending on their needs and established goals. It is often possible to trade certain properties to gain in performance, response times, scalability, reactiveness, or throughput in reasoning tasks. Agents may need to establish protocols that allow them to reach mutually agreeable terms with respect to these parameters [22]. Moreover, it is also key to establish control and monitoring mechanisms in order to assess the level of compliance to any agreement, in a trusted and transparent manner.

Streaming data privacy. Privacy is a major concern, especially concerning sensitive data, which can also be represented as continuous data streams (e.g., wearable sensor data). Agents may require to enforce privacy protection guarantees, implemented through anonymization or obfuscation. Access constraints may also be applied by agents to other agents, not only for data reuse, but also for processing and publication tasks, as detailed above. With current norms as the GDPR, agents may also act on behalf of data owners and stakeholders, making sure that their rights are respected.

3 OPPORTUNITIES

Stream reasoning can benefit from multi-agent systems in many aspects related to the challenges previously described. This synergy may create interesting opportunities and open doors to new research areas. It is noteworthy that agents actually fit into the stream reasoning paradigm in a number of ways. First, they are already equipped with continuous semantics, i.e. the ability to natively handle data processing a continuous manner. While this is performed in general over incoming sets of events, this is conceptually similar to what is done in CEP (Complex Event Processing) [2], a special case of stream processing. The ability to plug stream reasoners into the processing core of an agent is a feature that would constitute a key building block for a Web of connected stream processing engines. As a second point, we may also mention the goal-oriented nature of intelligent agents. This characteristic feature matches the needs of stream reasoners, which are also directed by objectives described in terms of; processing efficiency (optimization), time and deadlines (e.g. usage of windows, reporting policies [15]), response time boundaries, and continuous query requirements. These elements, which in classical stream reasoners are specified in diverse forms, can be formulated explicitly as part of the agent goals. The third point is related to the intrinsically reactive nature of agents. Reactive answers to queries, deductive reasoning tasks, complex event processing operations, etc., are essential for any stream reasoner. Although in traditional stream reasoners these reactive responses take the form of (continuous) query answers, or (continuous) entailments [17], from the agent perspective they can be seen as interactions, which can potentially be exchanged among different agents, or clusters of agents.

A fourth point refers to the capability of agents to capture knowledge and incorporate it to their beliefs or conceptualizations of their context. In stream reasoning this typically takes the form of knowledge graphs, encoded through ontology models that provide a semantic layout. The incorporation of such models may open the door for a new wave of semantically-aware agents, which can use this knowledge to operate on the highly dynamic data that continuously arrives as input.

Nevertheless, there are also some aspects that will require additional effort, from a scientific point of view, in order to allow agents to implement stream reasoning approaches. First, stream reasoning agents will need to understand their environment, and provide interactions that are suitable for it. This environment, the Web, has specific characteristics and limitations, which will have and impact over the type of behaviors that the agent will develop. Governing protocols for the Web such as HTTP, or mechanisms like WebSockets are some examples of technological assumptions, although there are other fundamental aspects of the Web that agents must consider: asynchronous communication, notions of URIs as identifiers and resource de-referencing, resource linking, redirection [9], etc. Another key aspect is the data layer. Agents need to modulate their sensing capabilities, i.e. consider both traditional incoming messages, as well as incoming streams of data. While messages can be used as part of a negotiation protocol, or for metadata exchange, streams can be ingested in a continuous manner, in push or pull mode, and fed through different mechanisms. Stream reasoning agents may perform two fundamental actions over a stream: consume or produce it. A consumer agent will need to include the ability to discover, search, get access and connect to relevant streams, while the producer will have to incorporate publishing and access control mechanisms. Agents may evidently implement both profiles, and allow the creation of stream reasoning workflows. Multi-agent stream reasoners may form networks that may work collaboratively, driven through common goals and self-organization, to fulfill common tasks. This will allow the combination of reasoners of entirely different nature, which have so far only worked as stream processing silos, with little or no capacity for a coherent division of work [16].

4 BUILDING BLOCKS

The stream reasoning agents vision relies on the achievements and experience of both the multi-agent systems and stream processing research communities. The challenges and opportunities presented previously will be translated into a research agenda, using the results produced by these two research areas as building blocks.

Multi-agent systems have been successfully used for solving a wide range of tasks in several domains.Through different paradigms and strategies, agents are able to act/react considering their goals, expectations, previous knowledge, and their environment. They can rely on using intelligent algorithms, continuous learning, and knowledge management techniques [19] to achieve their goals, in a decentralized manner. Tied to the decentralized nature of multi-agent systems (MAS) is the incorporation of coordination and communication mechanisms over complex networks, allowing the exchange of information regardless of their physical location. We believe that the Web is the natural environment for such interactions, as it counts not only with wide adoption but also a set of standards for efficient data exchange and linking.

The Web, however, is extremely complex and heterogeneous, and requires overcoming semantic boundaries and incompatibilities. The *Semantic Web* [8] initiatives have addressed these concerns to a large extent, and while its original vision considered agents as primary actors for the generation and consumption of data on the Web, in practice they have been somehow neglected. Most implementations of the Semantic Web have focused on ontology modeling, reasoning engines, Linked Data, or RDF data querying, but have relegated agents to a marginal position. Nevertheless, there are examples of previous works that proposed fundamental contributions, especially regarding the use of ontologies for defining agent knowledge bases [25], the development of Semantic Web Services [28] for orchestration and negotiation, or the inclusion of reasoning in intelligent agent behavior [18].

Regarding stream processing, most of previous works have focused on optimization and sophisticated processing techniques, often disregarding cooperation, decentralization and orchestration aspects. A relevant type of stream reasoners, focused on stream processing over semantically enriched data has been developed under the name of RDF stream processing (RSP). RSP engines have been developed focusing on the processing aspects of RDF streams, including incremental reasoning, continuous querying and complex event processing [2, 6, 11, 23, 31]. Although these stream processors disregard to a certain degree the Web dimension, they provide core functionalities for stream reasoning that could later be added to intelligent and autonomous agents.

Regarding service interface for stream reasoners, we can mention examples such as the RSP Service Interface or the SLD Revolution framework [3], providing generic implementable programming APIs for RSP query engines. Related to these efforts, the publication of streams on the Web has been addressed recently by efforts like TripleWave [27] and WeSP [16], which allow the publication of these RDF streams so that they can be directly consumed or connected with applications that process them. Finally, the emergence of data stream vocabularies as Vocals [37], can be an first step towards the establishment of ontologies for self-describing streams managed by distributed multi-agent systems.

5 APPLICATION SCENARIOS

Elaborating on the presented *challenges, opportunities*, and *building blocks*, several application scenarios can be envisioned. In the IoT spectrum, more and more use-cases require processing continuous data streams in possibly unknown (or new) environments with a incomplete or outdated knowledge. For example, being able to *discover* and *reuse* information from data streams can be crucial for MAS operating in *crowd management* [35]. In large events (e.g., music festivals, concerts, gatherings [1]) distributed sensors are dynamically employed to monitor and analyze the people flow. Thanks to this understanding, more UAV-cameras and new access point can be released in the area of the event, thus coping with the continuous evolution of the situation [29]. On a larger scale, smart cities can also benefit from stream reasoning agents, e.g. using data stream publication to regulate aerial or territorial vehicle traffic [39]. In such a scenario, negotiating over publicly available data is crucial (especially considering the eventuality of sudden changes). Moreover, *negotiation* and *interpretation* are crucial in agent coordination, which is especially relevant for user interaction and NLP (e.g., chatbots). Finally, concerning robots (e.g., UAVs) interactions and coordination, *privacy* and *security* concerns are outstanding when dealing with data streams [20]. Hence, reasoning on open perspectives for use cases such as unmanned fire-fighting and disaster recovery is still an open challenge.

6 ROAD-MAP

Stream reasoning addresses fundamental problems linked to a challenging combination of data velocity, volume, and variety on the Web. However, there is still a need for intelligent stream reasoning systems that can collaborate, self-organize, and exchange streams on the Web to fulfill common and individual goals. Multi-agent systems natively incorporate several of the fundamental principles that will boost this novel area of research. We propose the following research agenda for the stream reasoning agents vision:

Vocabularies. The inclusion of vocabularies and ontologies for streaming data description, discovery, provenance and exchange within agent systems is of particular importance. Based on initiatives like Vocals, or domain-specific ones in IoT, healthcare, etc., this will enable the publication and reuse of data streams on the Web, while adding the inherent advantages of an agent-based ecosystem.

Federation. A first step towards cooperation in a decentralized fashion is to enable the federation of different types of streaming engines. Federation of stream reasoners has been studied to a limited extent, often from a top-down perspective. The stream reasoning agents paradigm will go beyond this and focus on the capability of agents to re-organize according to common-established goals [33], and the optimization of the reasoning tasks, considering strengths and weaknesses of each engine. This is not straightforward considering the volatile and dynamic nature of streams which may impact the assumptions of the federation during the processing stages.

Negotiation. Largely neglected in stream reasoning so far, this aspect refers to the possibility of stream reasoners to negotiate [22] on data stream availability, processing sharing, set-up of common agreements, output delivery, etc. These considerations are critical in many scenarios where data stream access is subject to technical and cost-based constraints, and where scalability is at stake. It is often preferable to gain in responsiveness, even if accuracy or completeness levels are reduced. Finding acceptable trade-offs can be a matter of negotiation among stream reasoning agents, while having well-defined goals and expectations.

Cooperation. Beyond federation, a natural step forward will be to allow stream reasoning agents to explore cooperation schemes [30], where data and processing offerings can take place in a fully decentralized environment deployed on the Web. This can expand far beyond institutional or enterprise boundaries, and include interactions with the Internet of Things, with agents acting on behalf of connected-self data sources. For example, cooperation networks can be established in order to perform sensor crowd-sourcing tasks, where all the data provision conditions and access are regulated by user-proxy agents, and even some simple reasoning tasks can be delegated to leaf nodes in the cooperation network.

Privacy. Ensuring privacy protection, using different approaches spanning from obfuscation to anonymity guarantees, will be expected in any stream reasoning agents system. The existence of malicious agents in this environment cannot be disregarded, and therefore the necessary trust and transparency mechanisms need to be put into place. The usage of blockchain and distributed ledger technologies may also offer instruments that can contribute to achieve these goals [12].

REFERENCES

[1] Lamia Alabdulkarim, Wafa Alrajhi, and Ebtesam Aloboud. 2016. Urban Analytics in Crowd Management in the Context of Hajj. In *International Conference on Social Computing and Social Media*. Springer, 249–257.

[2] Darko Anicic, Paul Fodor, Sebastian Rudolph, and Nenad Stojanovic. 2011. EP-SPARQL: a unified language for event processing and stream reasoning. In *WWW*. 635–644.

[3] Marco Balduini, Emanuele Della Valle, and Riccardo Tommasini. 2017. SLD Revolution: A Cheaper, Faster yet more Accurate Streaming Linked Data Framework. In *RSP*. 1–15.

[4] Davide Barbieri, Daniele Braga, Stefano Ceri, Emanuele Della Valle, Yi Huang, Volker Tresp, Achim Rettinger, and Hendrik Wermser. 2010. Deductive and inductive stream reasoning for semantic social media analytics. *IEEE Intelligent Systems* 25, 6 (2010), 32–41.

[5] Davide Francesco Barbieri, Daniele Braga, Stefano Ceri, Emanuele Della Valle, and Michael Grossniklaus. 2010. C-SPARQL: a Continuous Query Language for RDF Data Streams. *Int. J. Semantic Computing* 4, 1 (2010), 3–25.

[6] Davide Francesco Barbieri, Daniele Braga, Stefano Ceri, Emanuele Della Valle, and Michael Grossniklaus. 2010. C-sparql: a continuous query language for rdf data streams. *Intl. J. Semantic Computing* 4, 01 (2010), 3–25.

[7] Harald Beck, Minh Dao-Tran, Thomas Eiter, and Michael Fink. 2015. LARS: A logic-based framework for analyzing reasoning over streams. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*. AAAI Press, 1431–1438.

[8] Tim Berners-Lee, James Hendler, and Ora Lassila. 2001. The semantic web. *Scientific american* 284, 5 (2001).

[9] Christian Bizer, Tom Heath, and Tim Berners-Lee. 2009. Linked data-the story so far. *Semantic services, interoperability and web applications: emerging concepts* (2009), 205–227.

[10] Jean-Paul Calbimonte, Davide Calvaresi, and Michael Schumacher. 2017. Multi-agent Interactions on the Web Through Linked Data Notifications. In *Multi-Agent Systems and Agreement Technologies*. Springer, 44–53.

[11] Jean-Paul Calbimonte, Hoyoung Jeung, Oscar Corcho, and Karl Aberer. 2012. Enabling query technologies for the semantic sensor web. *Int. J. Semantic Web Inf. Syst.* 8 (2012), 43–63.

[12] Davide Calvaresi, Alevtina Dubovitskaya, Jean Paul Calbimonte, Kuldar Taveter, and Michael Schumacher. 2018. Multi-agent systems and blockchain: Results from a systematic literature review. In *International Conference on Practical Applications of Agents and Multi-Agent Systems*. Springer, 110–126.

[13] Michael Compton, Payam Barnaghi, Luis Bermudez, RaúL GarcíA-Castro, Oscar Corcho, Simon Cox, John Graybeal, Manfred Hauswirth, Cory Henson, Arthur Herzog, 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 (2012), 25–32.

[14] Emanuele Della Valle, Stefano Ceri, Frank Van Harmelen, and Dieter Fensel. 2009. It's a streaming world! Reasoning upon rapidly changing information. *IEEE Intelligent Syst.* 6 (2009), 83–89.

[15] Daniele Dell'Aglio, Emanuele Della Valle, Jean-Paul Calbimonte, and Oscar Corcho. 2014. RSP-QL semantics: a unifying query model to explain heterogeneity of RDF stream processing systems. *International Journal on Semantic Web and Information Systems (IJSWIS)* 10, 4 (2014), 17–44.

[16] Daniele Dell'Aglio, Danh Le Phuoc, Anh Le-Tuan, Muhammad Intizar Ali, and Jean-Paul Calbimonte. 2017. On a Web of Data Streams. In *Proc. ISWC Workshop on Decentralizing the Semantic Web DeSemWeb 2017*.

[17] Daniele Dell'Aglio, Emanuele Della Valle, Frank van Harmelen, and Abraham Bernstein. 2017. Stream reasoning: A survey and outlook. *Data Science* Preprint (2017), 1–25.

[18] Mark d'Inverno et al. 2004. The dMARS architecture: A specification of the distributed multi-agent reasoning system. *Autonomous Agents and Multi-Agent Systems* 9, 1 (2004), 5–53.

[19] Jacques Ferber. 1999. *Multi-agent systems: an introduction to distributed artificial intelligence*. Vol. 1. Addison-Wesley Reading.

[20] Eduardo Castelló Ferrer. 2018. The blockchain: a new framework for robotic swarm systems. In *Proceedings of the Future Technologies Conference*. Springer, 1037–1058.

[21] Sebastian Kaebisch and Takuki Kamiya. 2017. Web of things (WoT) thing description. *First Public Working Draft, W3C* (2017).

[22] Sarit Kraus. 1997. Negotiation and cooperation in multi-agent environments. *Artificial intelligence* 94, 1-2 (1997), 79–97.

[23] Danh Le-Phuoc, Minh Dao-Tran, Josiane Xavier Parreira, and Manfred Hauswirth. 2011. A native and adaptive approach for unified processing of linked streams and linked data. In *ISWC*. 370–388.

[24] Maxime Lefrançois. 2017. Interopérabilité sémantique libérale pour les services et les objets. In *Actes de la 17ème conférence Extraction et Gestion des Connaissances (EGC'17), Grenoble, France*.

[25] Sean Luke, Lee Spector, David Rager, and James Hendler. 1997. Ontology-based web agents. In *Proceedings of the first international conference on Autonomous agents*. ACM, 59–66.

[26] Fadi Maali, John Erickson, and Phil Archer. 2014. Data catalog vocabulary (DCAT). *W3C Recommendation* 16 (2014).

[27] Andrea Mauri, Jean-Paul Calbimonte, Daniele Dell'Aglio, Marco Balduini, Marco Brambilla, Emanuele Della Valle, and Karl Aberer. 2016. TripleWave: Spreading RDF Streams on the Web. In *ISWC*. 140–149.

[28] Sheila A McIlraith, Tran Cao Son, and Honglei Zeng. 2001. Semantic web services. *IEEE intelligent systems* 16, 2 (2001), 46–53.

[29] Yazan Mualla, Amro Najjar, Stéphane Galland, Christophe Nicolle, Igor Tchappi, Ansar-Ul-Haque Yasar, and Kary Främling. 2019. Between the Megalopolis and the Deep Blue Sky: Challenges of Transport with UAVs in Future Smart Cities. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*. International Foundation for Autonomous Agents and Multiagent Systems.

[30] Reza Olfati-Saber, J Alex Fax, and Richard M Murray. 2007. Consensus and cooperation in networked multi-agent systems. *Proc. IEEE* 95, 1 (2007), 215–233.

[31] Xiangnan Ren, Olivier Curé, Li Ke, Jeremy Lhez, Badre Belabbess, Tendry Randriamalala, Yufan Zheng, and Gabriel Kepeklian. 2017. Strider: an adaptive, inference-enabled distributed RDF stream processing engine. *Proceedings of the VLDB Endowment* 10, 12 (2017), 1905–1908.

[32] Yuan Ren and Jeff Z Pan. 2011. Optimising ontology stream reasoning with truth maintenance system. In *CIKM*. 831–836.

[33] Giovanna Di Marzo Serugendo, Marie-Pierre Gleizes, and Anthony Karageorgos. 2005. Self-organization in multi-agent systems. *The Knowledge Engineering Review* 20, 2 (2005), 165–189.

[34] Nigel Shadbolt, Tim Berners-Lee, and Wendy Hall. 2006. The semantic web revisited. *IEEE intelligent systems* 21, 3 (2006), 96–101.

[35] Deepak Sharma, Amol P Bhondekar, AK Shukla, and C Ghanshyam. 2016. A review on technological advancements in crowd management. *Journal of Ambient Intelligence and Humanized Computing* (2016), 1–11.

[36] John Soldatos, Nikos Kefalakis, Manfred Hauswirth, Martin Serrano, Jean-Paul Calbimonte, Mehdi Riahi, Karl Aberer, Prem Prakash Jayaraman, Arkady Zaslavsky, Ivana Podnar Žarko, et al. 2015. Openiot: Open source internet-of-things in the cloud. In *Interoperability and open-source solutions for the internet of things*. Springer, 13–25.

[37] Riccardo Tommasini, Yehia Abo Sedira, Daniele Dell'Aglio, Marco Balduini, Muhammad Intizar Ali, Danh Le Phuoc, Emanuele Della Valle, and Jean-Paul Calbimonte. 2018. VoCaLS: Vocabulary and Catalog of Linked Streams. In *International Semantic Web Conference*. Springer, 256–272.

[38] Riccardo Tommasini and Emanuele Della Valle. 2017. Challenges & Opportunities of RSP-QL Implementations. In *Web Stream Processing workshop (WSP 2017) at ISWC*. 48–57.

[39] Christoph Torens, Florian-M. Adolf, and Lukas Goormann. 2014. Certification and Software Verification Considerations for Autonomous Unmanned Aircraft. *Journal of Aerospace Computing, Information and Communication* 11, 10 (2014), 649–664.

[40] Mark D Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E Bourne, et al. 2016. The FAIR Guiding Principles for scientific data management and stewardship. *Scientific data* 3 (2016).