Formalizing Stream Reasoning for a Decentralized Semantic Web

Mathijs van Noort*

Ghent University, Technologiepark-Zwijnaarde 126, 9052 Ghent, Belgium
`Mathijs.vanNoort@UGent.be`

Abstract. Decentralized storage of data is gaining increased attention as a means to preserve privacy and ownership over personal data. Simultaneously, the share of streaming data on the Web and other applications, e.g. Internet of Things, continues to grow. The large, uncoordinated amount of data streams within these applications requires methods that can coordinate them, especially when a central authority is lacking. We aim to perform said coordination via stream reasoning, using rules and facts to combine and derive information. Decentralized networks, however, present new problems for stream reasoning not yet (fully) addressed in the literature. This includes added expressivity for network heterogeneity, cross-storage referencing and schema variation, out-of-order arrival of data and variance in the representation of time. We aim to propose theoretical solutions that address challenges on temporal expressivity within the network, on out-of-order processing and on the alignment of temporal ontologies. Ultimately, this research aims to provide a solid formal basis for the processing of unbounded streaming data across different data vaults.

Keywords: Decentralized Data Processing · Stream Reasoning · Formal Language

1 Introduction

Recent years have seen increased attention towards decentralized systems, in particular a possible decentralization of the existing Web ecosystem [28]. By decentralizing data and putting access control in the hands of the end-user, rising topics, such as data silos, data ethics and data legislation, can be tackled from a fundamentally different angle. SOLID [29] positions itself as a set of specifications for design of personal data vaults and their content, in which each user, i.e. the vault owner, determines who gets access to which data. Standards set by SOLID are based on existing Semantic Web standards and make explicit use of Linked Data. By providing an alternative for the current centralization of data, the SOLID initiative opens up the possibility of a shift in data storage and processing from centralized to decentralized. Both governmental and industrial actors have expressed interested in this novel technology and have committed to development of large-scale applications including healthcare and employment domains [7, 10].

---

* Early Stage Ph.D.

Following these developments, derivation of meaningful insights requires the processing of data distributed over a large amount of (relatively) small, decentralized data vaults. This poses distinct technical challenges. When end-users have control over data in their own data vault, those users will naturally publish and handle their data in different ways, interpret the same information differently and express that information in ways that make the most sense for them personally. To deal with this heterogeneity in data, Solid uses Semantic Web specifications to make this difference in meaning explicit. The contents of a SOLID vault consists of documents of Linked Data for which ontologies are used to express the metadata of these documents and data, indicating how it should be interpreted by processing applications. Additionally, one vault can refer to information inside other vaults, leading to cross-storage referencing of information. As in a decentralized environment each vault owner can use their preferred ontologies to express these semantics, Semantic Web reasoning techniques are required to align these schemas and thus meaningfully combine data from multiple data pods.

Within this ecosystem of decentralized data vaults, we turn our attention towards one type of data, being streaming data. Streaming data is the continuous flow of real-time information. The continuous nature of this stream of data implies collected data will never be 'complete' and is unbounded, thus at no point in time can one claim to have collected all data. Streaming data has become omnipresent. As expressed by IDC, the portion of real-time data is expected to climb to 30% of the global datasphere by 2025 [15]. It can be found in Internet of Things (IoT), on social media platforms, stock markets, video games and many others. In most uses cases, streaming data is characterized by a high velocity, where data arrival oftentimes invalidates previous data. This rapid turnover demands frameworks that can express the temporal aspects of the data and techniques that can process said data as efficiently as possible. These challenges related to streaming data are not tied to specific applications, but are inherent to data streams and processing thereof. The research field that addresses how to reason upon heterogeneous data streams, as well as their representation, abstraction and integration in applications, is referred to as *stream reasoning* (SR) [8, 9, 14].

In a decentralized environment, dealing with streaming data becomes even more challenging. As discussed earlier, users retain a high degree of freedom in their choice to store, publish and semantically annotate their data. Specifically for streaming data, users may employ different notions and structures when it comes to expressing temporal information. A multitude of Temporal Logics (TLs) and temporal processing techniques, called stream reasoning, have emerged to handle various needs of ontology designers and users with respect to temporal expressivity [9]. In a decentralized environment, these different ontology designs are allowed to coexist and are not subjected to design constraints issued by a central authority. In the example given by Figure 1, vault B uses an ontology (depicted as graph) different from the ones used in vaults A and C. In order to facilitate information exchange within peer-to-peer networks, these different temporal semantics need to be aligned to avoid conflicts or miscommunication. As the temporal semantics employed within a data vault can change over time, (re)alignment also needs to be flexible enough to handle such changes. We identify this alignment of different temporal semantics across data vaults as our first challenge, the challenge of (A) *time-oriented schema alignment*.

In order to perform such time-oriented schema alignment, we aim to design sets of mappings between different temporal schemas. These mappings must therefore take into account the time at which information holds, as well as the 'source' of the information, i.e. the vault. To facilitate these mappings, we aim to identify a logic framework that can capture various different temporal schemas (and hence semantics). This framework thus needs to have a high expressivity. Our second challenge is therefore the search for (B) *a formal logic framework for decentralized stream reasoning*. This includes exploring the applicability of existing frameworks to streaming data in a decentralized system, as well as the development of a novel framework that aims to better suit the new system's needs if no existing framework covers all needs.

Lastly, the velocity of data arrival in the personal vault may differ drastically between streams due to factors such as data vault transmission rates, server location with respect to the data source, bandwidth availability and others. Due to different arrival frequencies, data may not arrive in a chronological order in the vault. An example is given in Figure 1, where vaults A and B each start an identical stopwatch simultaneously and transmit its advancement to a third vault C. A delay on connection A-C leads to double the transmission time compared to connection B-C, leading to an order of arrival that differs from the true order of events. Applications performing real-time processing of the data in vault C, will thus be faced with out-of-order arrival of data. Additional techniques are required to restore order to the incoming data or new techniques are needed. Our challenge consists of designing (C) *algorithms for out-of-order processing* that make optimal use of the decentralized architecture.

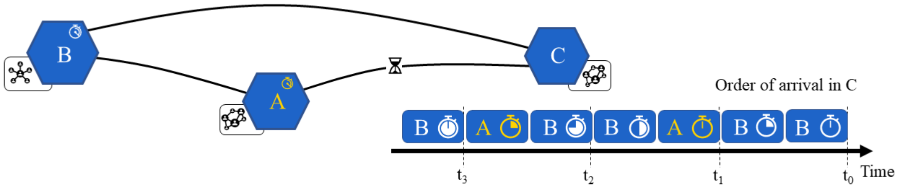

Fig. 1. An example decentralized network with out-of-order arrival in C when A and B start transmitting stopwatch times simultaneously to C, with a delay over the connection A-C. In addition, the ontologies of A and C differ from B's.

To solve the above elaborated three challenges, my Ph.D. research will mainly focus on theoretic, logic-based aspects of reasoning on streaming data in an environment of decentralized data storage – in short decentralized stream reasoning[1]. I will thereby aim to (i) design a logic framework that will serve as a formal basis for a high-level declarative language for decentralized SR, (ii) provide a set of algorithms to tackle challenges in stream processing specific to the decentralized setting, (iii) provide proofs of the theoretical properties of proposed framework and algorithms, and (iv) explore incremental and caching approaches for real-time continuous schema alignment. With the proper expressivity and techniques at

---

[1] 'decentralized' refers to the manner in which data is stored. No preliminary assumptions are made regarding the execution of the reasoning itself.

their disposal, developers are equipped to maximally leverage the potential of the decentralized Web. The abstract nature of the research aims to provide a theoretic fundament upon which virtually any application domain can function.

2 State of the Art

Based on the identified challenges, three areas of related research are elaborated.

2.1 Formalism for Decentralized Stream Reasoning

Multiple research domains have developed methods for handling streaming data. The ambition of an abstract basis for reasoning, and in particular SR, has been pursued in multiple directions. DLs have been extended to accommodate for SR, resulting in Linear Temporal Logic (LTL), Metric Temporal Logic (MTL) and others, at times grouped under the term Temporal Description Logics (TDL) [12]. The temporal expressivity for each of these TDLs differs due to different choices in semantics. Recent advancements aim to further extend the expressivity. In the work of Gutiérrez-Basulto et al. [12], aspects of LTL and MTL are taken to provide a layer of abstraction that merges qualitative constraints (e.g. event A happens *before/after* event B) and an explicit quantification of time to obtain a more potent logic for temporal reasoning. Temporal Logics have also been incorporated in rule languages, for example DatalogMTL [30] adds temporal operators from MTL into Datalog, enabling DatalogMTL to address challenges that require temporal reasoning, while taking advantage of the recursive properties of Datalog. A lack of formalisation of languages for SR has been pointed out by Beck et al. [5]. With LARS, these authors propose a model-based semantics that is closely linked with the theory of Answer Set Programming (ASP) [5]. The Semantic Web query language SPARQL has been extended in the form of C-SPARQL and CQELS [4, 17]. Both of these extensions introduce windowing mechanisms to SPARQL. Via LARS, these window mechanisms are given a formal basis. A formal basis allows us to analyse theoretical properties, such as model checking and satisfiability. LARS programs can also be seen as a generalization of answer set programs [5]. The close link between LARS and ASP semantics allows for cross-analysis and comparison between the two frameworks. Eiter et al. [11] adapted the LARS framework to suit networks with distributed decision-making components. The networks with distributed reasoning considered by Eiter et al. [11] share similarities to the networks of decentralized data storage considered in Section 1. The methods for stratification of the reasoning process can serve as inspiration for similar processing methods in the decentralized storage network.

2.2 Decentralized Time Semantics

As mentioned in Section 1, out-of-order arrival of data raises questions on timing of processing, especially when delay times are variable across streams and hard to predict. Results can be invalidated by arrival of 'late' data or may require incremental updates. Akidau et al. [1] provide a layer of abstraction for streaming and batch data as part of the Dataflow model. Analogously, Apache Flink merges batch processing, continuous streams and real-time analytics under a single stream processing model [6]. The Open Data Fabric, introduced in Mikhtoniuk and Yalcin [20] makes explicit use of Apache Flink to demonstrate its potential as decentralized exchange protocol for structured data. Both Dataflow and Apache Flink rely on watermarks to monitor divergence of event time and processing time and to (re)introduce order in case of out-of-order arrival. The added complexity of using watermarks is minimal in settings with limited distributed computing and homogeneous delays on data arrival. Watermarks, however, are used in conjunction with windowing [2]. The use of windowing, however, prevents continuous semantics. Current SR languages do not yet offer solutions that are able to preserve continuous semantics [26]. Lastly, LDQL provides semantics that allow link traversal in a Web of Linked Data for querying [13]. Semantics of link patterns allow for evaluation of data distributed over multiple documents. It is therefore suited to capture knowledge distributed over data vaults.

Internet of Things (IoT) systems face similar problems as systems on a decentralized Semantic Web. The issue of timing alignment noted in Marinier et al. [19] and Tu et al. [27] shows a resemblance to the out-of-order processing problem we aim to address. From the perspective of a processing agent, disparity in data generation rate of the various IoT sources and network heterogeneity on the decentralized Semantic Web induce similar patterns of out-of-order data. The decentralized Semantic Web, however, requires peer-to-peer communication, compared to IoT systems where there is often one ‘authoritative’ processing agent (or multiple). This authoritative figure can impose some uniformity, whereas in decentralized environment no such figure exists. This lack of a single authority has implications for the ways in which problems, such as alignment, can be solved. Nonetheless, research in the field of IoT that addresses timing alignment may serve as a starting point for solutions in a decentralized SR contexts. Our interest goes out to the ISDI architecture proposed by Tu et al. [27], as it addresses both out-of-order data as well as data integration from multiple sources, akin to our problem of real-time schema alignment.

2.3 Time-Oriented Schema Alignment

In order to express temporal information, either in absolute terms or in relative to other pieces of information, existing ontologies have seen temporal extensions [21, 32] and newly developed ontologies have incorporated temporal aspects directly [18]. Specifically for OWL, Abir et al. [32]introduces methodology for creating and updating ontology as well as ontology instances. Furthermore, Krieg-Brückner et al. [16] details how Generic Ontology Design Patterns (GODPs) can be employed to introduce time into previously atemporal ontologies. Work on schema alignment specifically geared towards time and temporal concepts were not discovered during exploratory research. Alternatively, recent survey works on ontology alignment, such as Ardjani et al. [3], may serve as a starting point for the development of tailored alignment techniques.

3 Problem Statement

The current state-of-the-art on SR focuses mainly on centralized systems. The overall objective of my research is to provide a formal basis for SR in decentralized systems. The research questions below each aim to support different aspects of this decentralized SR.

RQ-I **Declarative language for stream reasoning in a decentralized environment.** Decentralized data storage complicates tasks of data retrieval and processing that befall query engines and reasoning agents. Our attention goes out to three factors; (a) variation in data due to different schemas, (b) heterogeneity between vaults in data generation speed and (c) cross-storage referencing of information. Starting from logic languages and frameworks underpinning existing declarative languages, can the ones addressing these factors separately be combined into a single logic language that address all three factors, satisfying challenge (B) considered above? Can a declarative language be constructed that is sufficiently high-level, to hide system complexity and process details from the end-user? Lastly, can the semantics be formalized to ensure a uniform interpretation of the language in case of a multi-agent network?

RQ-II **Decentralized time semantics and out-of-order processing.** Given challenge (C) of processing streaming data in a decentralized environment, what heuristics can be *constructed* in order to process data that arrives out-of-order? How can results be updated when data arrives ‘late’, or how do previous results need to be invalidated?

RQ-III **Formal proof of the functionality of language and algorithms designed.** Is the language for decentralised time semantics sound and complete? If not, do there exist fragments of the language that meet these criteria? Can the correctness of the algorithm(s) be guaranteed? What are the theoretical time and memory complexities of the algorithms? In contrast to the other RQ’s, RQ-III does not address a specific challenge stated in Section 1, but rather serves to verify the validity of the answers to RQ-I and RQ-II, thus implicitly supporting challenges (B) and (C).

RQ-IV **Time-oriented schema alignment** As data providers are free to choose in what schema their data is stored, a reasoning process over a decentralized network of data providers is inevitably confronted with different semantics and representations of time. In the interest of the time-oriented schema alignment of challenge (A), how can different time schemas be aligned in real-time fashion, with minimal delay towards network users? Can the most common schemas on the Web be aligned to allow for reasoning on all available data? If a temporal logic framework can serve as a layer of abstraction over different schemas, how do different schemas then map to said abstraction layer?

4 Research Methodology

To ensure our envisioned new formalism builds upon the existing research, I aim to analyze the existing work on formalisms for SR, as elaborated in Section 2, via the format of a survey paper. This survey paper should provide insight into the expressivity of current formalisms, as well as missing links with respect to their usability in a decentralized environment.

The research done in the context of this survey paper aims to serve as a gateway towards tackling the challenge of RQ-I. The construction of the new framework is divided in multiple components. First, a fitting selection of temporal semantics must be assessed. Point-based, interval-based, answer set and other semantics each have distinct properties, degrees of complexity and influence data on a different level. The choice of temporal operators within those semantics also strongly influences the kind of temporal relations that can be expressed. The first task thus encompasses selection of semantics that meets the expressivity needs covered in RQ-I. Second, the formalism needs to capture chosen semantics in the appropriate mathematical structures, i.e. terms, formulae, models etc. Lastly, entailment regimes for resulting formulae need to be defined intuitively yet unambiguously. In these three components, a high degree of mathematical rigor is crucial. It serves to formally document semantics, as to allow interoperability between agents, and to enable the proofs pledged by RQ-III.

Regarding the design of algorithms for out-of-order processing and decentralized time semantics (RQ-II), the primary focus goes out to the development of heuristics. Due to the high volatility of the data and their high degree of unpredictability, rapid approximations made by heuristics are preferable from a practical perspective. The algorithms should make optimal use of the semantics defined through RQ-I. By exploiting the increased expressivity, the algorithms should aim to strike a balance between the accuracy of the answer and the computational power it requires. As the increased volatility and the data distribution encumber the direct application of watermarks, an adaption or generalization of watermarking offer an opportunity for novel out-of-order processing techniques. The design of the algorithms is performed in alternating fashion with the correctness proofs of RQ-III, in a ‘check-and-improve’ iterative fashion.

On RQ-III, existing results on soundness and completeness of languages are leveraged maximally in order to prove the sound- and completeness of the decentralized time semantics given by RQ-I. This includes the direct application of existing theoretical results, the translation of (fragments of) the new language into a language with documented results and the recuperation of proof methodology. An analogous methodology will be applied for the proofs regarding the algorithms considered in RQ-II. In cases where no existing results on soundness and completeness can be leveraged, conventional methods in the analysis of correctness and complexity will be enlisted. The ultimate approach for RQ-III is evidently heavily reliant on the outcomes of RQ-I and RQ-II.

As schema alignment is a broad research topic, we aim to limit the topic of schema alignment in this thesis to alignment of temporal concepts. We aim to identify possible areas of conflict, e.g. discrete vs. continuous time and interval- vs. point-based semantics, and aim to utilize the framework from RQ-I to construct mappings between the various semantics. By using said framework as a ‘turntable’ between semantics, discrepancies in expressivity can be exposed and investigated.

5 Evaluation plan

The results of this research will primarily be evaluated through the analysis of theoretical properties. In essence, the evaluation of RQ-I and RQ-II is in part incorporated in the RQ-III. The proofs on soundness and completeness of the proposed framework (RQ-I) aim to support the suitability of the framework and to verify the quality. In order to assess usability of the framework – or fragments thereof –, we will determine the decidability (or undecidability) of the system.

Analogously for the proofs concerning the algorithms of RQ-II, the (partial) correctness assesses their usability. The time and memory complexities of the algorithms serve as metrics to gauge their competitiveness w.r.t. existing algorithms for centralized systems as well as their applicability in real-life use cases. In addition to these theoretical results, the algorithms will be implemented as part of larger use cases in an e-health context, using the datasets and ontologies provided by the DAHCC project [24].

The techniques for real-time schema alignment will be evaluated in terms of data processing time and required computational power. Possible benchmarks to consider include those adopted by the Ontology Alignment Evaluation Initiative [31], in which case the exact benchmark will be identified among those available at the time of evaluation, taking into account the relative niche of the application domain (streaming data in Linked Data networks with decentralized data storage) Geared toward the SOLID environment, the SolidBench [25] benchmark simulates a social network environment in which we can evaluate the schema alignment techniques. The suitability of each of the benchmarks above will need to to be investigated further.

6 Preliminary Results

I commenced the research of my Ph.D. on decentralized stream reasoning in September 2022. The results thus far are hence limited. They can be divided into two main areas. On the one hand, preparatory work on the survey paper has led to a temporary selection of 13 papers between the years 2018 and 2022 that fall within our scope of SR formalisms. As the field of SR is relatively young [22], inclusion of sources from the domains of ASP and IoT is taken into consideration. Among the sources gathered to date, there are none who address the topic of decentralized data storage explicitly. Expectation is to progress towards publication in a fitting peer-reviewed journal by the end of 2023.

On the other hand, some exploratory work has focused on evaluation of temporal operators. I have obtained first results on defining relations between various temporal logic operators currently in use in the literature. These results focus on rewriting operators expressing statements such as '... happened (at least) once before', '... will always be true in the future' ($\diamondminus$ and $\boxplus$ resp.) and others in function of each other without reliance on a negation operator. As a result, this work aims to provide a minimal set of temporal operators that retains temporal expressivity in negation-less logic frameworks (or fragments thereof) compared to frameworks with negation. These negation-less frameworks can be used to model languages such as RDF and Datalog, which have only limited support for negation (e.g. stratification). The results also streamline new proofs as they only need to cover a smaller set of operators. This scopes within RQ-I as a means of exploratory research, as well as within RQ-III as provision of potential auxiliary lemmas for the intended proofs. These results are currently being bundled for submission to the Conference on Principles of Knowledge Representation and Reasoning [23].

7 Conclusions

In the above, I outlined the research plan for my Ph.D. where I aim to provide a formal basis for SR in a Semantic Web environment with decentralized data storage. Via the four research questions, I identified specific challenges within the environment and delimited the manner in which this thesis aims to address these challenges. The state-of-the-art presented in Section 2 covers relevant research, listing work from within the fields of Semantic Web and stream reasoning as well as several works, taken from other research fields, that address similar challenges.

As this thesis focuses heavily on the theoretical aspects, future work will be on the implementation and empirical evaluation of the outcome of this research. This also includes enlisting the theory in more use cases, preferably covering a wide variety of application domains.

Acknowledgements. This research is funded by the FWO Project FRACTION (Nr. G086822N). I thank my supervisors Pieter Bonte (UGent) and Femke Ongenae (UGent) for the feedback and comments on the paper. I thank Maarten Vandenbrande (UGent) for the shared discussions.

References

[1] Akidau, T., et al.: The dataflow model: a practical approach to balancing correctness, latency, and cost in massive-scale, unbounded, out-of-order data processing. Proceedings of the VLDB Endowment **8**(12), 1792–1803 (2015)

[2] Akidau, T., et al.: Streaming systems: the what, where, when, and how of large-scale data processing. O'Reilly Media, Inc. (2018)

[3] Ardjani, F., et al.n: Ontology-alignment techniques: Survey and analysis. International Journal of Modern Education & Computer Science **7**(11) (2015)

[4] Barbieri, D., et al.: C-SPARQL: A continuous query language for RDF data streams. International Journal of Semantic Computing **4** (2010)

[5] Beck, H., et al.: LARS: A Logic-based framework for analytic reasoning over streams. Artificial Intelligence **261**, 16–70 (2018)

[6] Carbone, P., et al.: Apache Flink™: Stream and batch processing in a single engine. The Bulletin of the Technical Committee on Data Engineering **38**(4) (2015)

[7] Cellan-Jones, R.: NHS data: Can web creator Sir Tim Berners-Lee fix it? BBC News (2020), https://www.bbc.com/news/technology-54871705

[8] Della Valle, E., et al.: It's a streaming world! reasoning upon rapidly changing information. IEEE Intelligent Systems **24**(6), 83–89 (2009)

[9] Dell'Aglio, D., et al.: Stream reasoning: A survey and outlook. Data Science **1**(1-2), 59–83 (2017)

[10] Digitaal Vlaanderen: Solid ecosysteem. https://overheid.vlaanderen.be/informatie-vlaanderen/ontdek-onze-producten-en-diensten/solid-ecosysteem (2020)

[11] Eiter, T., et al.: A distributed approach to LARS stream reasoning (system paper). Theory and Practice of Logic Programming **19**, 974–989 (2019)

[12] Gutierrez-Basulto, V., et al.: On metric temporal description logics. In: ECAI 2016: 22nd European Conference on Artificial Intelligence. vol. 285, pp. 837–845. IOS Press (2016)

[13] Hartig, O., Pérez, J.: LDQL: A query language for the web of linked data. Journal of Web Semantics **41**, 9–29 (2016)

[14] Hirzel, M., et al.: Stream processing languages in the big data era. ACM SIGMOD Record **47**(2), 29–40 (2018)

[15] IDC: IDC Futurescape outlines the impact ”digital supremacy” will have on enterprise transformation and the IT industry (2019), https://www.idc.com/getdoc.jsp?containerId=prUS45613519

[16] Krieg-Brückner, B., et al.: Generic ontology design patterns: Roles and change over time. arXiv preprint arXiv:2011.09353 (2020)

[17] Le-Phuoc, D., et al.: A native and adaptive approach for unified processing of linked streams and linked data. In: ISWC 2011: 10th International Semantic Web Conference, Proceedings, Part I 10. pp. 370–388 (2011)

[18] Madkour, M., et al.: Temporal data representation, normalization, extraction, and reasoning: A review from clinical domain. Computer methods and programs in biomedicine **128**, 52–68 (2016)

[19] Marinier, P., et al.: Maintaining time alignment with multiple uplink carriers (Jan 13 2015), US Patent 8,934,459

[20] Mikhtoniuk, S., Yalcin, O.N.: Open Data Fabric: A decentralized data exchange and transformation protocol with complete reproducibility and provenance. arXiv e-prints pp. arXiv–2111 (2021)

[21] Milea, V., et al.: tOWL: a temporal web ontology language. IEEE Transactions on Systems, Man, and Cybernetics, Part B **42**(1), 268–281 (2011)

[22] Pacenza, F.: Reasoning in Highly Reactive Environments. arXiv e-prints arXiv:1909.08260 (Sep 2019)

[23] Principles of Knowledge Representation and Reasoning, Inc.: KR2023 - KR Conference (2023), https://kr.org/KR2023/

[24] Steenwinckel, B., et al.: Data Analytics For Health and Connected Care: Ontology, Knowledge Graph and Applications. In: 16th EAI Pervasive Healthcare Conference. Springer (2022), https://dahcc.idlab.ugent.be

[25] Taelman, R.: Solidbench. https://github.com/SolidBench/SolidBench.js (2023)

[26] Tommasini, R., Sakr, S., Balduini, M., Della Valle, E.: Tutorial: An outlook to declarative languages for big steaming data (2019)

[27] Tu, D.Q., et al.: IoT streaming data integration from multiple sources. Computing **102**(10), 2299–2329 (2020)

[28] Verborgh, R.: Re-decentralizing the Web, for good this time. In: Linking the World’s Information: A Collection of Essays on the Work of Sir Tim Berners-Lee (2022)

[29] W3C Solid Community Group of the W3C: Solid technical reports (2020), https://solidproject.org/TR/

[30] Wałega, P.A., et al.: Reasoning over streaming data in metric temporal datalog. Proceedings of the AAAI Conference on Artificial Intelligence **33**(1), 3092–3099 (2019)

[31] Zamazal, O.: A survey of ontology benchmarks for semantic web ontology tools. International Journal on Semantic Web and Information Systems **16**(1), 47–68 (2020)

[32] Zekri, A., et al.: τOWL: A systematic approach to temporal versioning of semantic web ontologies. Journal on Data Semantics **5**(3), 141–163 (2016)