Semantic Web 0 (0) 1
IOS Press

A Holistic View over Ontologies for Streaming Linked Data

Pieter Bonte [a,*], Femke Ongenae [a] and Riccardo Tommasini [b]
[a] *Ghent University - imec, Ghent, Belgium*
E-mails: pieter.bonte@ugent.be, femke.ongenae@ugent.be
[b] *INSA Lyon, LIRIS, France,*
E-mail: riccardo.tommasini@insa-lyon.fr

Abstract. Applied research and prototypes constitute an important part of the initiative around Stream Reasoning (SR) research. From Social Media analytics to the monitoring of IoT streams, the SR community worked hard on designing working prototypes, query languages, and benchmarks. Applied work that uses stream reasoners in practice often requires a data modeling effort. For this purpose, RDF Stream Processing (RSP) engines often rely on OWL 2 ontologies. Although the literature on Knowledge Representation (KR) of time-varying data is extensive, a survey investigating KR for Streaming Linked Data is still missing.

In this paper, we describe an overview of the most prominent ontologies used within SR applications and compare their data modeling and KR capabilities for Streaming Linked Data. We discuss these ontologies using three complementary KR views, i.e. viewing the streams as Web resources, a view on the structure of the stream, and a view on the modeling of the events in the streams themselves. For each view, we propose an analysis framework to facilitate fair comparison and in-depth analysis of the survey ontologies.

Keywords: Stream Reasoning, RDF Stream Processing, Web Stream Processing, Knowledge Representation

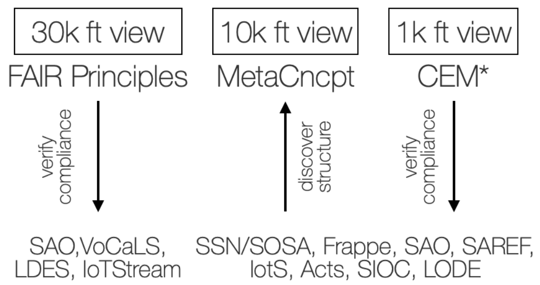

Fig. 1. The paper's contributions. [F]indable, [A]ccessible, [I]nteroperable, [R]eusable; [C]o[NC]e[PT]ualization; [C]ommon [E]vent [M]odel.

1. Introduction

In recent years, the Semantic Web community has witnessed a growing interest in streaming data for application domains that combine the presence of data variety (i.e., highly heterogeneous data sources) with the need to process data as soon as possible and before they are no longer useful (Data Velocity). Examples of such application domains include Smart Cities, Industry 4.0, and Social Media Analytics. Stream Reasoning (SR) is a research initiative that combines Semantic Web with Stream Processing technologies to the extent of addressing the aforementioned challenges at the same time. SR counts several research outcomes that span across Continuous Querying, Incremental Reasoning, and Complex Event Recognition [9]. RDF Stream Processing (RSP) is a subarea of SR that focuses on the processing of RDF Streams [20]. In particular, the research activities around RSP, include a growing number of applied research works due to the availability of working prototypes, benchmarks, and libraries [14] that, in turn, spawn research on Streaming Linked Data [21, 24].

While data streams become more available on the Web, the community started discussing best practices to publish data streams in an interoperable manner. To

*Corresponding author. E-mail: pieter.bonte@ugent.be.

this extent, the FAIR data initiative is promising. Indeed, Tommasini et al. reinterpreted some of the steps of the linked data lifecycle to answer the question "*how can we make (streaming) data Findable, Accessible, Interoperable, and Reusable (FAIR)* [21]?".

Tommasini et al. consider several resources published under the SR umbrella. A number of works emerged that show how to access and process data streams on the Web. Yet, little has been done regarding the data modeling and knowledge representation efforts that Streaming Linked Data applications entail. Even though a number of domain-specific ontologies have been used in Streaming Linked Data applications.

In this paper, we dig deeper into this claim surveying the related literature and isolate such efforts. In particular, we investigated research papers that apply SR/RSP as a solution. Like in similar works, we select the papers in a systematic manner, defining inclusion criteria and filtering methods. From these selected papers, we extracted the used ontologies to model the data streams. We study such ontologies from three perspectives: (i) A **thirty-thousand foot view**, which observes streams as Web resources and surveys existing practices for data modeling and KR for data streams. This view starts from the FAIR principles [26] and verifies the compliance of several ontologies under survey. (ii) A **ten-thousand foot view**, which further zooms in on the streams and investigates the structure of stream reasoning ontologies. This novel framework is a meta-conceptualization that results from a bottom-up analysis of the stream reasoning ontologies, guided by a stream processing conceptualization. (iii) A **thousand foot view**, which narrows further down and observes the data stream internals, focusing on the items that populate the streams. This analysis framework is inspired by the Common Event Model [25] and verifies the compliance of the inner parts of the stream representation.

We summarize the papers contribution in Figure 1. For the 30k ft view, we follow a top-down approach. Our analysis framework is based on the FAIR principles [26] as they were adapted in [21]. Thus, we verify the compliance of the selected ontologies. For the 10k ft view, we proceed inductively from the selected ontologies to extrapolate a meta-conceptualization. Such process is guided by grounded concepts that constitute SR's theoretical footprint. Finally, for the 1k ft vie we proceed once again top-down using the Common Event Model [25], which was originally designed for multimedia application, yet whose semantics align with SR requirements.

Outline: Section 2 introduces the necessary background to understand the content of the paper. In Section 3 we introduce the ontologies that are being investigated. Sections 4, 5, and 6 present the three views from higher to lower. Section 7 details the related work and Section 8 concludes the paper.

2. Preliminaries

In this section, we present the fundamental notions needed to understand the content of the paper. In particular, we present the survey methodology and the Streaming Linked Data lifecycle.

2.1. Survey Methodology

To conduct our survey, we followed the guidelines of the systematic mapping research method [5], which has been already used successfully for surveys in the semantic web [15]. In particular, our investigation aims at answering the following research question (RQ):

RQ1 *What characterizes the knowledge representation efforts for managing heterogeneous data that are streaming or highly dynamic?*

The integration of heterogeneous data is a significant part of Semantic Web Research. In addition, RQ1 includes two main components, i.e., *Streaming/Highly Dynamic Data* and *knowledge representation*. The former relates to application domains like the Internet of Things or Social Media Analytics (but also financial analysis, smart cities, and cluster management). The latter is central in applications that deal with complex information needs. Together, they point to contributions from the Stream Reasoning community and, in particular, to Streaming Linked Data. Indeed, under the SR initiative several engines, query languages, and benchmarks were proposed to address Streaming Linked Data use cases.

To collect relevant studies, we initially conducted a keyword-based search on Google Scholar, the IEEE Xplore, and the ScienceDirect and investigated their citations to retrieve further interesting studies. We used the following keywords to retrieve 620 papers:

- Stream Reasoning
- RDF Stream Processing
- Streaming Linked Data
- Incremental Reasoning
- Ontology AND Streaming/Dynamic

– Ontology AND Event
– Observation AND Ontology

The next steps of our collection apply a number of filters to reduce the number of papers and narrow the analysis. To this extent, we identified different inclusion criteria (IC) indicated below. Notably, IC1-4 are based on the papers' metadata, while IC5 and IC6 are content-based.

IC1 papers should be written in English
IC2 papers should be peer-reviewed
IC3 papers should be published in the last 10 years,
IC4 papers should have at least 10 citations.
IC5 papers should *apply* a SR/RSP solution to process data streams,
IC6 papers should present/reuse a domain-specific ontology to model the data in the processed streams,

Like in [15], we apply *Metadata-based* filtering to screen papers screening their title, abstract, and publication venue and, then, we apply *Content-based* filtering step drilling down to the papers introduction, conclusion and if needed, the full text. Finally, we proceeded with an enrichment step (aka *snowballing*) which aim at expanding the relevant papers based on investigating their citations and related work. Especially for papers proposing Streaming Linked Data engines, it was very beneficial to investigate their citations as it revealed many use case papers.

In conclusion, our analysis identified 32 papers from which we extracted 10 ontologies. One last step of our analysis was dividing the ontologies into two groups. The first group addresses streaming linked data from a publication/discovery standpoint. Given the abstract view, we name such group 30kft view. The second group looks a streaming linked data from a processing standpoint, which is a lower level of abstraction. Therefore, we name this group the 10kft view. We also notice that within the latter group there is an even lower abstraction point of view which we name the 1k ft view and concerns the representation of data points within the streams. Figure 2 visualises the selection process.

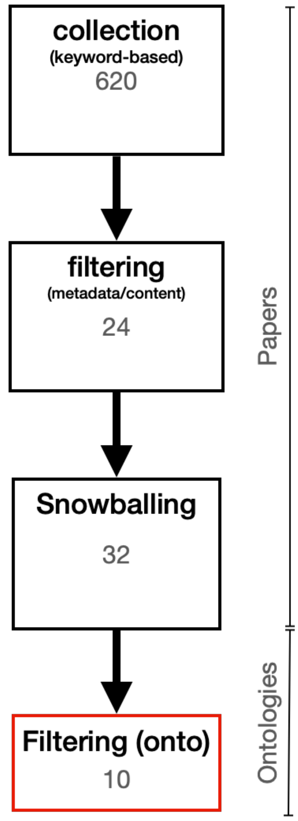

Fig. 2. Collection and Filtering methodology visualized.

2.2. *Streaming Linked Data Lifecycle*

The Streaming Linked Data Lifecycle proposes a number of guidelines on how to manage data streams on the Web. Figure 3 depicts the full life-cycle and highlights the *Model* and *Describe* step which both require a knowledge representation effort. The *Model* step takes care of modeling the content of the stream using a certain ontology-based knowledge representation, while the *Describe* step focuses on describing the stream itself as a Web resource. The latter aligns with the 30k ft view, while the former aligns with the 10k and 1k ft view. Each of these steps requires the use of stream-specific ontologies and (rich) metadata. While the other steps are out of scope for this paper, it is worth mentioning that Step (0) is about naming Web Streams using appropriate URIs; Step (3) is about streaming data conversion; Step (4) is about serving data using protocols that enable continuous data access (e.g., WebSockets), and Step (5) relates to Web Stream Processing.

3. Selected Works

This section details the selected SR ontologies that will be investigated using the proposed 30k, 10k, or 1k ft view.

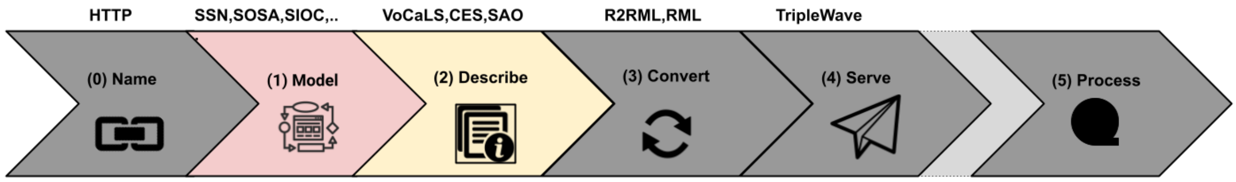

Fig. 3. Streaming Linked Data Life-Cycle from [21]

Ontology	Prefix	Relevant Classes	Relevant Properties
OWL-Time	time	`TemporalEntity`, `TimeInstant`, `TimeInterval`	`inXSDDateTimeStamp`, `hasTime`
PROV-O	prov	`Activity`, `Event`	`atTime`
DCAT	dcat	`Dataset`	
Event Ontology	eo	`Event`	

Table 1
Summary of Foundational Ontologies

3.1. *Foundational Ontologies*

We first describe four general ontologies that are frequently imported in the SR ontologies we will discuss later. Moreover, we highlight parts of their conceptualizations that are relevant to understand the content of the paper.

OWL Time[1] is an ontology that captures temporal concepts. It is extensively used to describe the temporal properties of Web resources. OWL Time models both temporal intervals and instants. Its conceptualization includes, but is not limited to, dates, temporal entities, and Allen's Algebra Relations.

PROV-O[2] captures the PROV data model using OWL2. The ontology aims at enabling provenance information exchange across systems.

DCAT[3] is an RDF vocabulary designed to foster interoperability among data web-published catalogs. It focuses on describing how data catalogs and datasets are accessible and distributed.

Event Ontology[4] is an OWL ontology originally designed in the context of the Music Ontology by the Centre for Digital Music. The ontology was intended to describe performances, compositions, recordings, or sound generation. Nevertheless, its generality fostered its adoption making EO the most used event ontology in the Linked Data community [16].

[1] https://www.w3.org/TR/owl-time/
[2] https://www.w3.org/TR/prov-o/
[3] https://www.w3.org/TR/vocab-dcat-2/
[4] http://motools.sourceforge.net/event/event.html

Ontology	Prefix	30kft	10kft	1kft
VoCALS	vocals	✓	✓⁻	
LDES	ldes	✓		
SSN/SOSA	ssn/sosa		✓	✓⁻
SAREF	saref		✓	✓⁻
IoT Stream	iots	✓	✓	✓⁻
SIOC	sioc		✓	✓
LODE	lode		✓	✓⁻
ActS	acts		✓	✓
Frappe	frp		✓	✓
SAO/CES	sao/ces	✓	✓	✓

Table 2
Ontology for Stream Reasoning: Summary. (✓: supported, ✓⁻: partly supported)

3.2. *Ontologies for Stream Reasoning*

When surveying the literature, we found that the following ontologies are being used for the description and modeling of streaming data as Web resources:

The **Vocabulary for Cataloging Linked Streams (VoCaLS)** is an ontology [22] that aims at fostering the interoperability between data streams and streaming services on the web [22]. It consists of three modules for 1) publishing of streaming data following the Linked Data principles, 2) description of the streaming services that process the streams, and 3) tracking the provenance of stream processing [22].

The **Stream Annotation Ontology (SAO)** allows publishing derived data about IoT streams. It is designed to represent both raw and aggregated data. The vocabulary allows to describe the aggregation transformations in depth. SAO relies on PROV-O to track the aggregation provenance and OWL-Time for the temporal annotations [13].

The **Complex Event Ontology (CES)** extends OWL-S[5] to support automated discovery and integration of sensor streams. It was designed to describe event services and requests, therefore it can be used to annotate streaming services. However, there is no distinction between streams publisher and consumers. Provenance tracking is possible at the level of transformation by distinguishing primitive and complex event services. Notably, CES was designed to be used in combination with SAO and, thus, we consider them together in our analysis [11].

Linked Data Event Stream (LDES)[6] defines a collection of immutable objects that evolve over time, describing both historical and real-time updates. *LDES* uses the TREE specification[7] for the modeling of the collections and data fragmentation purposes when the size of the collections becomes too big for a single HTTP response. TREE defines a collection of objects that adhere to a certain SHACL shape, and how these collections can be fragmented and interlinked using multi-dimensional HTTP pagination [24].

IoT Stream a vocabulary for the annotation of (IoT) streams. It extends the SOSA ontology (see below) with the notion of Streams, Events and Analytics that can be extracted from the streams [10].

Furthermore, we additionally identified the following prominent ontologies used in RSP applied research and will investigate their structure and internals when used as a knowledge representation in stream reasoning applications:

The **Semantic Sensor Network (SSN)**[8] is the W3C recommendation to describe sensors, platforms, devices, and observations [18].

The **Sensor Observation Sampling Actuator**[9] **(SOSA)** ontology is the result of the community attempt to rewrite SSN to the extent of making the ontology more usable. The ontology integrates many rewriting proposals and ultimately reduces the ontological commitment of SSN by selecting a core module relevant for most IoT applications. It is a modular ontology design, where SSN can be seen as an extension of SOSA.

The **Smart Applications REFerence ontology**[10] **(SAREF)** aims at enabling interoperability between different IoT providers. It is similar to SOSA/SSN but provides specific classes for sensors and observations (called *Device*s and *Measurement*s), in comparison with SSN, which is very generic. SAREF thus has various extensions tailored for specific domains.

The **Linked Open Descriptions of Event (LODE)** is an RDFS vocabulary that aims at unifying existing event ontologies, such as the Event Ontology. LODE represents only *facts* using the 4W framework, i.e., *What, When, Where* and *Who* [16].

Frappe is a vocabulary for spatio-temporal streaming data analytics. Frappe borrows its conceptualization from the domain of photography. It represents the world as a sequence of frames. Events occur within a spatio-temporal context. To represent the spatial context Frappe uses three classes, i.e., Grid, Cell, and Place, and models time using the OWL Time ontology [2].

The **Semantically-Interlinked Online Communities (SIOC)** describes the information that online communities (e.g., wikis, weblogs, social networks, etc.) have about their structure and online community content [6].

The **Activity Streams 2.0 (ActS)**[11] vocabulary includes classes and properties to describe past, present and future activities. The vocabulary consists of (i) a core that generalizes the structure of an activity, and (ii) an extended module that includes properties that cover specific types of activities common to many social web application systems.

All surveyed ontologies, their prefixes and which views they cover are summarized in Table 2.

4. Thirty-Thousand Foot View: Web Streams

In this section, we present the thirty-thousand foot view for Streaming Linked Data. At this height, we observe data streams as Web resources, i.e., the fundamental building blocks of the World Wide Web, and we focus on their metadata, governance, and provenance. Only four of the ten selected ontologies have the no-

[5] http://citypulse.insight-centre.org/ontology/ces/
[6] https://w3id.org/ldes/specification
[7] https://w3id.org/tree/specification
[8] https://www.w3.org/TR/vocab-ssn/
[9] https://www.w3.org/2015/spatial/wiki/SOSA_Ontology
[10] https://saref.etsi.org/core/v3.1.1/
[11] https://www.w3.org/TR/activitystreams-vocabulary/

tion of data streams as Web resources, the others are not included in this discussion. These four ontologies include VoCALS, SAO/CES, LDES and IoTStream.

4.1. Analysis Framework

We analyze the selected ontologies that model streams as Web resources using the FAIR Principles [26] summarized below:

Findable. (F1) Data should be assigned unique and persistent identifiers, e.g., DOI or URIs. (F2) Data should be assigned metadata that includes descriptive information, data quality, and context. (F3) Metadata should explicitly name the persistent identifier since they often come in a separate file. (F4) Identifiers and metadata should be indexed or searchable.

Accessible. (A1) Data and metadata should be accessible via (a) free, (b) open-sourced, and (c) standard communication protocols, e.g., HTTP or FTP. Nonetheless, authorization and authentication are possible. (A2) Metadata should be accessible even when data is no longer available.

Interoperable. (I1) Data and metadata must be written using formal languages and shared vocabularies that are accessible to a broad audience. (I2) Such vocabularies should also fulfill FAIR principles. (I3) Data and metadata should use qualified references to other (meta-)data.

Reusable. (R1) Data should adopt an explicit license for access and usage. (R2) Data provenance should be documented and accessible. (R3) Data and metadata should comply with community standards.

Tommasini et al. reinterpreted the FAIR Principles for Streaming Data Management [21]. During our analysis, we build upon these preliminary adaptations.

Definition 1 introduces the notion of Web stream, which is a prerequisite for identifying streams on the Web.

Definition 1. *A Web data stream is an unbounded ordered collection of pairs (o, i), where o is a Web resource, and i is metadata that can be used to establish an ordering relation, e.g., a timestamp.*

4.2. Discussion

We now analyze the selected ontologies, w.r.t. the FAIR data principles. Table 3 summarizes the analysis. Notably, we did not include the evidence for I2 and I3 in the table. Regarding I2, none of the ontologies respect I2, as they all rely on at least one non-FAIR import. Regarding I3, all the ontologies support it, since they all reuse concepts from other vocabularies.

Identity (F1) According to Tommasini et al., for uniquely identifying data streams, it is necessary to consider them as Web resources. Three out of four of the selected ontologies, i.e., *VoCaS*, *LDES*, and *IoTStream*, introduce a similar conceptualisation. More specifically, *VoCaLS* includes the notion of `voc:Stream` specifically to represent an unbounded dataset on the Web. Similarly, *LDES* introduces the notion of `ldes:EventStream` as an append-only collection of immutable elements, and assigns to it a retention policy. Finally, Elsaleh et al. include in their IoT Stream ontology the notion of `iot:IoTStream`.

Metadata (F2) According to the FAIR initiative, metadata should be generous and extensive, including *descriptive* and *contextual* information about the data, as well as indications of data quality. In these regards, *VoCALS* does not allow to include any information on data quality but limits its support to descriptive information about the resources, e.g., name and owner, and contextual information, e.g., the vocabulary used to annotate the stream content. *SAO/CES* supports all three metadata annotations. It is worth noticing the presence of specific classes and properties for annotating data quality by extending *QOI*[12]. *LDES* explicitly supports only contextual metadata as it directly relies on the TREE specification. Finally, *IoTStream* also supports all three types of metadata, including classes and properties for describing the stream, the quality of the streamed data, and contextual resources, e.g., services.

RDF (I1) To support interoperability, the FAIR initiative suggests using community standards. In the context of streaming data, this immediately refers to RDF Streams. Only *VoCaLS* explicitly includes the RDF Stream conceptualization, while *LDES* on the other hand, assumes the consumption of RDF data.

Service (A1) The FAIR prescription for serving data and metadata relies on standard protocols. While on the Web this usually means HTTP, it does not directly apply to streaming data that call for specific protocols. Except *LDES*, which inherits the HTTP access assumption from *TREE*, the other ontologies include a specific abstraction that aims

[12]https://mobcom.ecs.hs-osnabrueck.de/cp_quality/.

Ontology	Identity (F1)	Metadata (F2)	RDF Stream (I1)	Service (A1)	Descriptor (F3,A2)	LICENSE (R1)	Provenance (R2)
VoCALS	✓	DC	✓	✓	✓	Apache 2	✓
SAO/CES		DQC		✓		CC	✓
LDES	✓	C	≃		✓	CC	
IoTStream	✓	DQC		✓		CC	✓

Table 3

Summary of the thirty-thousand foot view analysis. Legend: ✓=supported, ≃=partially supported, [D]escriptive, Data [Q]uality, [C]ontextual, [C]reative[C]mmons

at generalizing the access to the streaming data, i.e., `voc:StreamEndpoint`, `iots:Service`, `ces:EventService`.

Descriptor (F3, A2) Directly related to the Service abstraction is the FAIR requirement for decoupling data and metadata documents. For streaming data, this notion was originally introduced by Barbieri et al. [3], who suggested to share the stream metadata in a separate document accessible via HTTP. Only *VoCALS* and *LDES* adopt this convention, fulfilling requirements F3 and A2.

License (R1) All the selected ontologies have an explicit license. *VoCaLS* and *LDES* explicitly suggest associating a license with the annotated data streams.

Provenance (R2) Finally, tracking the provenance of the shared data is an encouraged practice from the FAIR initiative. In these regards, all the ontologies, except for LDES, include dedicated classes and properties that allow to represent the analysis performed on the streaming data, i.e., `voc:Query` based on RSP-QL; CES `ces:EventPattern` for complex event recognition, `sao:StreamAnalysis` and `iots:Analytics` for continuous analysis of the data streams.

As Table 3 shows, we can conclude that the combination of VoCaLS with SOA/IoTStream allows to increase the FAIRness of the streams. It is important to note that every single ontology does not need to cover all aspects. It is possible to combine ontologies with different capabilities to obtain complete coverage.

5. Ten-Thousand Foot View: Streams' Structure

In this section, we present the Ten-thousand foot view of the surveyed streaming reasoning ontologies. This view focuses on the ontological level and analyses the nature and nurture of ontologies used for representing streaming data. Only eight of the ten selected ontologies describe concepts to represent the streaming data. These eight ontologies include SSN/SOSA, SAREF, IoTStream, SIOC, LODE, ActS, Frappe and SAO/CES. The other ontologies are not included in this discussion.

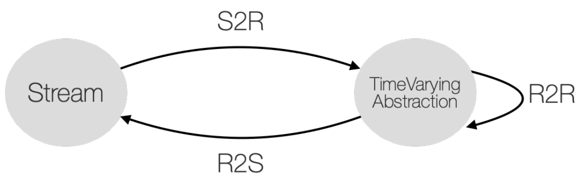

Fig. 4. Streaming Linked Data Abstractions

5.1. Analysis Framework

In the related literature [1, 8, 14], dynamic data are typically divided into two kinds of abstractions, i.e., unbounded time-ordered data a.k.a. *streams* and *time-varying* ones. Arasu et al. [1] introduced such data dichotomy to the extent of formalizing relational continuous queries Dell'Aglio et al. [7] extended it later on for RSP. In this work, we focus on Streaming Linked Data and, thus, RDF Streams (see Definition 2).

Definition 2. *An RDF Stream is a Web stream such that o is an RDF object, i.e., an RDF graph, a quad, or a triple, and $\tau \in T$ is a timestamp. An element (o,τ) is said to be* ***instantaneous****, to highlight its validity at a precise point in time τ.*

On the other hand, Time-varying abstractions represent the result of continuous computations (see Definition 3) and, as the term suggests, capture the changes that occur to data as a function of time. Definition 4 formalizes the notion and specializes for RDF data.

Definition 3. *Continuous Computations proceed under continuous semantics, i.e., they output an infinite stream while consuming one or more infinite streams as inputs.*

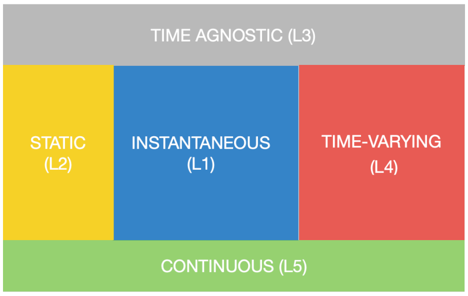

Fig. 5. Ontology Meta-Structure in SR.

Definition 4. *Time-Varying Abstractions (TVA) are functions that map the temporal domain to finite entity sets that relate with a given abstraction $T \rightarrow A$.*

In particular, a Time-Varying RDF Graph is a function $T \rightarrow G$, where T is the time domain and G is the set of possible RDF graphs.

Streaming Linked Data focus on query answering, i.e., continuous computations assume the form of Continuous Queries (CQ), which are a special class of queries that listen to updates and allow interested users to receive new results as soon as data becomes available.

Several extensions of SPARQL exist [9] to perform continuous queries over RDF Streams. RSP-QL [7] is a reference model that aims at unifying the formal semantics of existing SPARQL extension. It's abstraction can be found in Figure 4. A common aspect of these languages is the notion of windowing, which allows to perform stateful computation over a stream. Window Operators, a.k.a. Stream-to-Relation (S2R) operators, chunk the stream into finite portions where computations can terminate. Once windows are applied, operators that involve time-varying abstractions can be traced back to the original static case (R2R). Finally, a class of operator that transform back time-varying data into streams are called Relation-to-Stream (R2S). According to RSP-QL, a time-varying RDF Graph is the result of the application of a window operator over a stream.

Last but not least, **static** data co-exist with both streaming and time-varying ones. Indeed, stream enrichment with contextual static knowledge is a popular task in SR/RSP [14].

5.2. Discussion

In this section, we elicit the data dichotomy explained above to study the meta-conceptualization of the selected ontologies that model concepts that are in line with the meta-conceptualization described above. For this reason, LDES and VoCALS are not taken into account in this discussion.

Figure 5 shows a ten-thousand foot view of a stream reasoning ontology. An ontology used for SR typically consists of five levels, i.e., *L1* the *instantaneous level* identifies the part of the ontology that is directly associated with a temporal annotation. Entities of this kind occur in the stream. *L2* the *static level* of the ontology identifies those concepts that may have a temporal annotation, but that are assumed to not change while the continuous computation occurs. This level is relevant for the stream enrichment task [14]. For the sake of completeness, we also include a *time-agnostic level L3*, which identifies those ground terms that are independent of time. *L4* the time-varying level includes entities whose state evolves over time. Entities of this kind are typically the result of a continuous computation, e.g., an aggregation. Last but not least, we include the *continuous level L5* to identify those terms that combine other terms and return time-varying entities as a result of processing. Entities of this kind typically include continuous transformations or queries. Notably, due to the lack of space, we leave a deeper investigation of L5 as future work.

While the detailed analysis of the selected ontologies is presented below, it also summarized in Table 4.

Instantaneous (L1). There is a clear agreement between the IoT ontologies (SSN, SOSA, and IoT-Stream) which identify the `sosa:Observation` on their instantaneous level. SAREF's conceptualization is slightly different as `srf:Measurement` already includes the unit of measure. On the other hand, SAO/CES adopt a generic data item using the classes `sao:StreamData` and `sao:Point`. SIOC and ActS presents a small hierarchy of concepts, i.e., `sioc:Post`, `sioc:Item`, and `as:Activity` that capture the interaction with social networks (or general web interactions). Frappe and LODE adopt the concept of `Event`, which both align with the Event Ontology.

Static (L2). Also for the static level, the IoT ontologies share a similar conceptualization, i.e., `Device`, `Sensors`, and `Platforms` are entities that are assumed to be static when the analysis occur. Frappe's static part includes concepts for representing spatial information. ActS' static part is limited to the `as:Actor` class and its sub-classes. SIOC's static part relates to `Users` and `Spaces` that represent the population and logical location of online communi-

Ontology	Instantaneous (L1)	Static (L2)	Time Agnostic (L3)	Time-Varying (L4)	Continuous (L5)
SSN/SOSA	Observation, Result	Sensor, Platform,	ObservableProp., Measure		Procedure
SAREF	Measurement	Device	Property, UnitOfMeasure	State	Function
IoT Stream	Observation	Sensor, Service, Platform	Quality, Unit, QuantityKind	Event	Analytics
SIOC	Item,Post	User,Space	Role	Container	
LODE	Event				
ActS	Activity	Actor	Link	Collection	
Frappe	Event	Cell,Grid, Place		Pixel, Frame	Capture, Synthetize
SAO/CES	StreamData, Point	Service, Sensor		Segment, StreamEvent	Stream Analysis
VoCaLS		Stream, RDF Stream		SDS, TimeVaryingGraph	Task Operator

Table 4

Summary of the Ten-thousand foot view analysis.

ties. LODE does not include concepts at L2. Notably, VoCaLS includes `Stream` and `RDFStream` as static concepts. In fact, they are meant to represent streams as resources (to be continuously consumed).

Time Agnostic (L3). Neither Frappe nor SAO/CEO, which were originally designed for SR/RSP applications, directly include L3 concepts. On the other hand, the IoT ontologies include concepts that do not directly have a temporal dimension. Such entities are related to the properties observed from the sensors and the unit of measurement. While LODE does not include concepts at L3, SIOC and ActS respectively have only one, i.e., `sioc:Role` that represent the role of a `sioc:User` on a `sioc:Space` and `as:Link` that represent a generic connection between two resources.

Time Varying (L4) and Continuous (L5). Except for LODE all the selected ontologies present a time-varying part. On the other hand, L5 remains uncovered by LODE, SIOC, and ActS.

Interestingly, L4 is where the selected ontologies differ the most. SSN/SOSA distinguish between the `ssn:Result` of a `ssn:Procedure`, and the action taken after processing, i.e. a `ssn:Actuation`. SAREF represents continuous computations as `Functions` that aggregates `:Measurements` to modify a `srf:Device`'s `srf:State`. IoTStream's continuous part is called an `iots:Analytics` and produces `iots:Events` as time-varying entities. SAO/CES include the class `sao:StreamAnalysis` too. However, the result can be either a `sao:StreamEvent` or a `sao:Segment`, which is just a portion of the stream. Frappe includes a time-varying corresponding entity for both the static entities `frp:Grid` and `frp:Cell`, i.e., `frp:Frame` and `frp:Pixel`. As briefly mentioned, it also represents continuous entities, i.e., `frp:Capture` and `frp:Synthesize`. Last but not least, VoCaLS includes two entities inspired by RSP-QL [7], i.e., `TimeVaryingGraph` that represents the time-varying equivalent of an RDF Graph, and `SDS`, which is a collection of TimeVaryingGraphs. Moreover, VoCaLs explicitely mention continyous transformations, i.e., `Task` and `Operator`. The former is meant to generalize continuous queries, while the latter helps tracking provenance by representing the task internals.

We can see that most ontologies distribute the ontology complexity across different temporal levels, facilitating the alignment with SR applications.

6. Thousand Foot View: Streams' Content

In this section, we present our Thousand Foot View of Streaming Linked Data. This view focuses on the stream's internals. In particular, we study the notion of Ontology Kernel (see Definition 5), and how the selected ontologies implement it. We reuse the ontologies introduced in the Ten-thousand foot view. Only eight of the ten selected ontologies describe concepts to represent the stream's internals. These eight ontologies include SSN/SOSA, SAREF, IoTStream, SIOC, LODE, ActS, Frappe and SAO/CES. The other ontologies are not included in this discussion.

6.1. Analysis Framework

The Common Event Model (CEM) was initially proposed by Westermann and Jain for multimedia applications [25]. CEM is designed for historical event analytics. Thus, it does not relate to L4 and L5. When porting CEM to SR/RSP, we must reinterpret some aspects. Traditionally, data streams are characterized by a form of *punctuation* that allows streaming operators to iterate over an unbounded sequence of data [23]. In SR/RSP, punctuation relates to the stream shapes, e.g., Graph, Triple, Predicate, as well as with the notion of Event Types [9]. At the ontological level, this reflects on the levels conceptualization, especially L1. Thus, we introduce the following notion:

Definition 5. *An **Ontology Kernel** is the minimal set of classes and properties of a certain ontology used to represent the instantaneous level.*

In our analysis, we highlight the relation between the Kernel and the ontological layers presented in Section 5. Figure 7 visualizes the *Kernel*, highlighting CEM's dimensions and the ontological levels. CEM describes events' data according to the following dimensions:

Informational: the data and metadata that describe the event, e.g. the event type and other entities involved in the event.

Experiential: the data and metadata that link the event with the transporting media, e.g., images, sensors measurements, or audios snippets.

Spatial: data and metadata that describes *where* the event occurred. Spatial metadata are further organized in conceptual (e.g., a building), logical (e.g. an address), and physical definitions (e.g. coordinates).

Temporal: metadata that describe *when* the events occurred. Like the spatial dimension, the conceptual (e.g., time instants), logical (e.g., relative time), and physical (e.g. a UNIX timestamp) distinction applies. Moreover, CEM distinguishes between point-based and interval-based time semantics.

Structural: data and metadata about the event's structure, e.g., how they are aggregated and linked to each other. As RDF is being used to model the event, we identify four event structures based on query shapes, i.e., Stars, Cycles, Chains, and Trees, as visualized in Figure 6. Note that ontologies allow to model events using multiple shapes.

~~**Causal**~~: data and metadata that describe what caused the event and how. Notably, causality is a form of provenance that in SR is typically described at query level. Coherently with the assumption to leave processing as future work, we do not include it in the analysis.

~~**Composition**~~: Allows the event model to compose the events into a larger whole, e.g. a smoke and high temperature observation observed in the same room could be composed into a fire observation. We do not consider the composition or aggregation of events at the event modeling level, as SR allows to define compositions or aggregations at higher levels of abstraction [19].

6.2. Discussion

We now align each of the ontologies with the CEM: We distinguish the Informational and Experiential discussion over the two levels L1 and L2. The higher the level, the further away from the core. L1 is one property-link away from the core, e.g. a type assertion and linked entities, while L2 requires two hops, e.g. types of the linked entities of L2 or additional entities) We provide a summary of the analysis for the Informational and Experiential discussion in Table 5 and for the Spatial and Temporal discussion in Table 6.

Informational. On L1, the ontologies describe the types of the events. For the sensor ontologies (SSN, SOSA and IoTStream) the types of the events are `sosa:Observations`, with the extension of `iots:StreamObservation` for IoTStream. These ontologies are very generic, it is the responsibility of the user to further specify the *Observation* types, e.g. to add specific *Observation*s such as a *TemperatureObservation* to the ontology. SAREF describes `srf:Measurements` instead of `sosa:Observations` and already provides a number of specific types in a form of a hierarchy. Both SSN and SAREF specify a number of ontological restrictions that can be enforced by the reasoners, e.g. each `sosa:Observation` should be made by exactly one `sosa:Sensor`. SOSA is more lightweight as it does not contain any restrictions. SIOC describes `sioc:Items` and `sioc:Posts` as the event types, a shallow hierarchy, and no type restrictions

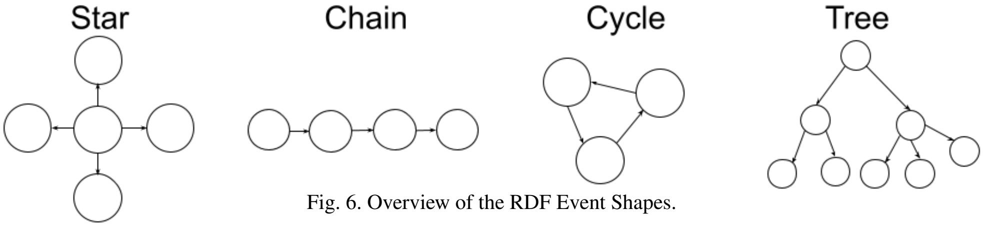

Fig. 6. Overview of the RDF Event Shapes.

	Level 1		**Level 2**	
Ontology	**Informational**	**Experiential**	**Informational**	**Experiential**
SSN	Observation + restrictions	Sensor values	Sensors, Systems, Properties. + restrictions.	None
SOSA	Observation	Sensor values	Same as SSN	None
IoT Stream	(Stream)Observation, Event	Sensor values	Same as SOSA, + IotStreams	None
SAREF core	Measurement + hierarchy + restrictions	Sensor values	Device, Property + hierarchy + restrictions	Device: model and manufacturer
SIOC	Item/Post + hierarchy(flat)	Post content: literal, attached file: URI.	User, UserGroup + hierarchy (flat)	Containers: size; Users: name and avatar
LODE	Event	None	Objects, Agents.	None
ActS	Activity + hierarchy	Name, content, summary	Objects, Links + hierarchy	Objects: name, content and summary.
Frappe	Event	event metadata	Place, Grid-Cell	Place: location metadata
SAO	Observation, StreamEvent	Sensor values, Stream analysis	Same as SSN, + StreamAnalysis	Stream Analysis: model parameters

Table 5
Overview of Ontology Kernel analysis for Informational and Experiential information.

are defined. In LODE, `lode:Event` is the central event type, no event hierarchies or type restrictions are included. `as:Activities` represent the main types in the ActS ontology. It defines a hierarchy of `as:Activities` and a small number of restrictions for some activity subtypes. Frappe imports `eo:Event` from the Event Ontology as event types with neither hierarchies nor restrictions. We see that L1 Informational type definitions are mostly very simple, except for SSN and SAREF. SSN has its lightweight version SOSA to make the modeling of the events more simple. The fact that the event description is rather simple in ontological complexity is in line with the Cascading Reasoning principle in SR that states that high-velocity streams should be processed with simple processing techniques, while once the streams have been filtered, more advanced processing can be performed using more expressive reasoning techniques [4]. Next to the event Types, L1 also links to the Entities that are involved in the event. On L2, informational data include the types of the L1 linked Entities which describe the Static level of the ontology. In particular, the IoT ontologies (SSN, SOSA, IoTStream and SAO) link the `sosa:Observations` to `sosa:Sensors` that made the observations and `sosa:ObservableProperties` that have been observed. IotStream has the additional `iots:IotStream` concept that `iots:StreamObservations` can belong to, while SAO links to the specific `sao:Stream Analysis` that was executed to extract the `iots:StreamEvent` from the `sosa:Observations`. SAREF links its `srf:Measurements` to `srf:Devices` (instead of *Sensors*) and the observed *Properties*. In SIOC, on an Informational L2, `sioc:Items` and `sioc:Posts` are linked to to the involved `sioc:Users` or `sioc:UserGroups`. In LODE, the `lode:Events` are linked to the involved `lode:Objects` and `lode:Actors` in a very generic way. `as:Activities` in ActS can be linked on an Informational L2 to the involved `as:Objects` and `as:Links`. In Frappe, the `eo:Events` are linked to `frp:Places` they are happening in. The ontological complexity of L2 is in line with L1, i.e., SSN and SAREF define restrictions, while SAREF, SIOC and ActS define hierarchies of concepts.

Note that many of the classes of Informational L1 align with the Instantaneous level of the Ten-Thousand Foot View even though these are two different ways of looking at the classes of the ontologies. In the previous, view we looked at the classes that had a temporal annotation, while in this view we look at the classes used for modeling the events. They align as the events themselves are what changes over time.

Experiential. On L1, experiential data are the event payload. The sensor ontologies (SSN, SOSA, IoT-Stream, SAO, and SAREF) describe sensor values. SIOC describes the post content and ActS describes the name, summary, and content (as html) of the activity. Frappe and LODE do not support experiential properties. On L2, experiential data are the static entities' metadata. SAREF allows its `srf:Devices`

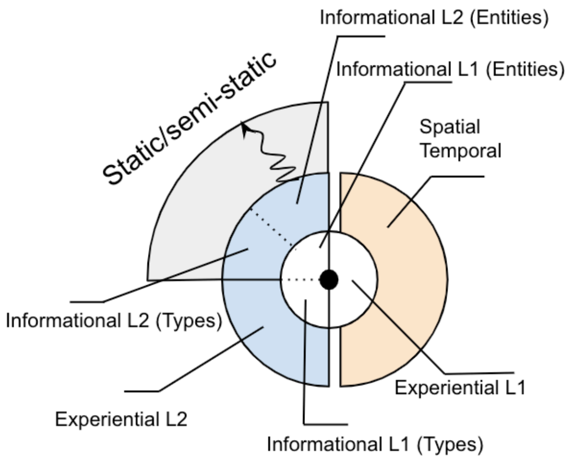

Fig. 7. Kernel Structure.

to have properties that can uniquely characterize it, namely its model and manufacturer. In SIOC `sioc:Users` and `sioc:UserGroups` can maintain metadata about their size, while users can have a name and avatar. In ActS, `as:Objects` can have all sorts of metadata such as name, content, and summary. All other ontologies do not support experiential L2 properties out of the box.

Temporal. SSN/SOSA defines two temporal concepts, i.e. `sosa:resultTime` and `sosa:phenomenonTime`. The data property `sosa:resultTime` has `xsd:dateTime` as range and provides point-semantics. The object property `sosa:phenomenonTime` is more expressive and allows to model both interval and point semantics through the use of `time:TemporalEntity`. In IotStream, the class `iots:StreamObservation` defines the interval of the window it belongs to using the data properties `iots:windowStart` and `iots:windowEnd` (with range `xsd:dateTimeStamp`). SAO allows the use of the TimeLine Ontology for both interval and point semantics for the extracted `soa:StreamEvents`. In SAREF, `srf:Measurements` can have point-semantics using the data property `srf:hasTimestamp` (with range `xsd: dateTime`), while `srf:Properties` can have both point and interval semantics using the object property `srf:hasTime` (with range `time:TemporalEntity`). In SIOC, `sioc:Posts` can be annotated using point-semantics using `dcterms:created` and `dcterms:modified` with a literal using ISO-8601 formatted date values. In LODE, the `lode:Events` can be timestamped both with point as interval semantics with the `lode:atTime` object property with `time:TemporalEntity` as domain that can model both point and interval semantics. In ActS, interval-based time semantics are supported using data properties `as:startTime` and `as:endTime` (with `xsd:dateTime` as range). In Frappe, `eo:Events` have point-based time semantics using the property `frp:time` with `time:Instant` as range.

Interestingly, we see that most ontology models rely on `xsd:dateTime` for point-semantics, while for interval-semantics, there does not seem to be a consensus. Some vocabularies chose to model their own intervals, e.g. `startTime` & `endTime`, while others rely on `time:TemporalEntity`.

Spatial. For the spatial definition, we make a distinction between physical, conceptual, and logical definitions. SSN, SOSA, and SAREF have no out-of-

Ontology	**Spatial**	**Temporal**
SSN	No support	Point (xsd:dateTime); Interval (time:TemporalEntity)
SOSA	Same as SSN	Same as SSN
IoT Stream	Physical locations (geo:Point).	Same as SSN Self defined Interval (xsd:dateTimeStamp)
SAREF core	No support	Point (xsd:dateTime) Interval (time:TemporalEntity)
SIOC	Logical	Point
LODE	Conceptual (dul:Place) Physical (geo:SpatialThing)	Point and interval (time:TemporalEntity).
ActS	Physical (lode:Place) Logical (lode:Place)	Self defined Interval (xsd:dateTime)
Frappe	Pyshical (geosparql:SpatialObject) Conceptual (geosparql:SpatialObject)	Point-semantics (time:Instant); Self defined Interval (xsd:dateTime).
SAO	Physical (geo:SpatialThing) Conceptual (geo:SpatialThing)	Same as SSN + Point and Interval (TimeLine Ontology)

Table 6

Overview of Ontology Kernel analysis for Spatial and Temporal information.

the-box support for spatial definitions. In IoTStream, the `iots:IotStreams` have physical locations defined through `geo:location` (with `geo:Point` as range). SOA allows modeling the location of Features of Interest that are being observed using `geo:SpatialThing`. In SIOC, logical locations are supported, i.e. `sioc:Sites` can be the location of an online community and a `sioc:Space` is defined as being a place where data resides. In LODE, `lode:Events` can have conceptual locations using `lode:atPlace` (with `dul:Place` as range) or physical locations using `lode:inSpace` (with `geo:SpatialThing` as range). In ActS, `as:Activities` can have both physical and logical definitions through the definition of the `as:Place` object. In Frappe, `eo:Events` can have both physical and conceptual locations defined through location (with `frp:Place` as range, which is a subclass of `geosparql:SpatialObject`). Note that `geosparql:SpatialObject` can define both physical and conceptual locations. We saw that physical spatial definitions typically rely on the `geo` and `geosparql` imported ontologies, while conceptual locations on `DUL` and `geosparql`.

Ontology	Star	Snowflake	Chain	Tree	Cycle
SSN	✓	✓	✓	✓	✓
SOSA	✓		✓		✓
IoT Stream	✓	✓	✓	✓	✓
SAREF core	✓	✓	✓	✓	✓
SIOC	✓		✓	✓	
LODE	✓		✓	✓	
ActS	✓	✓	✓		
Frappe	✓	✓	✓		
SAO	✓	✓	✓	✓	✓

Table 7

Structural Analysis vs Query Shapes

	Chain	**Star**	**Cycle**	**Tree**
L1: Informational(Type)		✓		✓
L1: Informational(Entity)	✓	✓	✓	✓
L1: Experiential		✓		✓
L2: Informational(Type)	✓			✓
L2: Informational(Entity)			✓	✓
L2: Experiential	✓			✓

Table 8

RDF shapes alignment with the kernel and ontology levels.

Structural. Figure 8 shows an example of the SOSA ontology, where both Chain, Stars, Cycles, and Trees can be used. However, we saw in the literature that the Star is most often used. The same holds for SSN, IoTStream, and SAREF. Other ontologies model both Chain, Stars, and Trees. However, the Star seems to be the best suited for streaming purposes. Indeed, when going up in ontology structure levels (e.g. Informational L2) data becomes more static, and as the event itself is typically kept limited in size, the more static data is not described in the event itself but linked through informational L1 (Entities).

Chains are not particularly useful as they only allow to move from the core of the kernel to outer level through Informational Entity relations. At the end of the chain there can optionally be only Informational Type or Experiential data, as these data end the chain. Cycles share the same faith, as they only allow to cycle through Informational Entity relations, without any Experiential or Type data, as these data end the cycle. Trees can model all data, but tend to describe unnecessary static data. Stars can model Informational L1, both the type of the event itself and the linked Entities, while describing the data in the Experiential L1, making it ideal for event modeling. Table 7 and 8 summarize the analysis.

7. Related Work

Dell'Aglio et al. [9] recently surveyed the state-of-the-art of stream reasoning research. They initially identified 9 requirements for a stream reasoning system to satisfy, then they analyzed the compliance of existing works to them. Although the authors discuss streaming annotation, which is comparable to our thirty-thousand foot view, they do not explicitly compare ontologies themselves.

Margara et al. [14] also surveyed solutions for stream reasoning and RDF stream processing. The focus of this survey was on comparing system capabilities and identifying limitations in terms of RDF stream processing. Although related to potential future work, we did not include *processing* in this current work. Thus, this survey can be seen as complementary.

In the context of the Semantic Web for the Internet of Things, the work of Szilagy et al. [17] is related. The authors discuss the advantages of semantic annotation for solving interoperability issues in the IoT domain. Then, they propose a specialized version of the Semantic Web stack for IoT. Although Szilagy et al. propose to compare four ontologies, including SSN, the comparison is not the main focus of their work. Moreover, the analysis's scope is limited to IoT and does not include ontologies like SIOC and LODE.

Finally, Gyrard et al. [12] describe a Linked Open Vocabulary (LOV) for IoT projects (LOV4IoT). LOV4-IoT identified existing IoT ontologies, re-engineered the vocabularies to make them interoperable, and cataloged them. However, they did not investigate each of the ontologies' capabilities for modeling data streams and LOV4IoT is limited to IoT applications.

8. Conclusion

In this paper, we surveyed the work on KR for Streaming Linked Data. In particular, we presented 1) a thirty-thousand foot view observing streams as Web resources, 2) a ten-thousand foot view that observes the nature and nurture of the ontologies for streaming data starting from a bottom-up approach and 3) a thousand foot view, which zooms further in and discusses how different ontologies model the events in the stream. Our analysis can be summarised as follows:

From **thirty-thousand foot**, most Stream description ontologies do not completely adhere to the FAIR principle. However, a combination of VoCALS and SAO/IoTStream fulfills most of the requirements. From **ten-thousand foot**, ontologies distributed their complexity alongside five time-related dimensions, i.e., Instantaneous (L1), Static (L2), Time Agnostic (L3), Time-Varying (L4), and Continuous (L5). The L4 is where most differences can be spotted. Most interestingly, ontologies explicitly designed for Streaming Linked Data ignore L3 and elaborate on L5. Finally, **from a thousand foot** we noticed that a *little semantic goes a ~~long~~ fast way*. Ontologies keep their *kernel* small under the assumption that the further away from the kernel, the more static the data. Additionally, while there is no consensus on how time is represented, a star-shaped event is the most prominent one.

As not all ontologies cover all aspects and different views, to be compliant with the Streaming Linked Data principles, a combination of SR ontologies is recommended.

As future work, we plan to extend the analysis to include a **five-hundred foot view** and a **hundred foot view** that respectively observe how (RDF) streams are serialized (data formats) and served (protocols).

References

[1] Arvind Arasu, Shivnath Babu, and Jennifer Widom. The CQL continuous query language: semantic foundations and query execution. *VLDB J.*, 2006.

[2] Marco Balduini and Emanuele Della Valle. Frappe: A vocabulary to represent heterogeneous spatio-temporal data to support visual analytics. In *ISWC*, 2015.

[3] Davide Francesco Barbieri and Emanuele Della Valle. A proposal for publishing data streams as linked data - A position paper. In *LDOW*, 2010.

[4] Pieter Bonte, Riccardo Tommasini, Emanuele Della Valle, Filip De Turck, and Femke Ongenae. Streaming massif: cascading reasoning for efficient processing of iot data streams. *Sensors*, 18(11):3832, 2018.

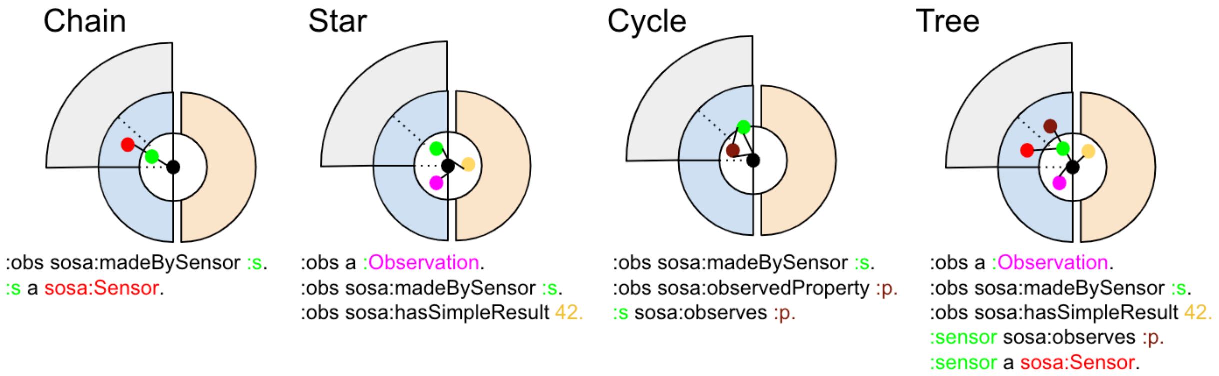

Fig. 8. Mapping of the RDF structures on the Event Kernel using the SOSA ontology.

[5] Pearl Brereton, Barbara A Kitchenham, David Budgen, Mark Turner, and Mohamed Khalil. Lessons from applying the systematic literature review process within the software engineering domain. *Journal of systems and software*, 80(4):571–583, 2007.

[6] John G. Breslin, Stefan Decker, Andreas Harth, and Uldis Bojars. SIOC: an approach to connect web-based communities. *Int. J. Web Based Communities*, 2(2):133–142, 2006.

[7] Daniele Dell'Aglio, Emanuele Della Valle, Jean-Paul Calbimonte, and Oscar Corcho. Rsp-ql semantics: A unifying query model to explain heterogeneity of rdf stream processing systems. *IJSWIS*, 10(4):17–44, 2014.

[8] Daniele Dell'Aglio, Emanuele Della Valle, Frank van Harmelen, and Abraham Bernstein. Stream reasoning: A survey and outlook. *Data Sci.*, 1(1-2):59–83, 2017.

[9] Daniele DellAglio, Emanuele Della Valle, Frank van Harmelen, and Abraham Bernstein. Stream reasoning: A survey and outlook. *Data Science*, 2017.

[10] Tarek Elsaleh, Shirin Enshaeifar, Roonak Rezvani, Sahr Thomas Acton, Valentinas Janeiko, and María Bermúdez-Edo.

[11] Feng Gao, Muhammad Intizar Ali, Edward Curry, and Alessandra Mileo. Automated discovery and integration of semantic urban data streams: The ACEIS middleware. *FGCS*, 2017.

[12] Amelie Gyrard, Christian Bonnet, Karima Boudaoud, and Martin Serrano. Lov4iot: A second life for ontology-based domain knowledge to build semantic web of things applications. In *FiCloud 2016*. IEEE Computer Society, 2016.

[13] Sefki Kolozali, María Bermúdez-Edo, Daniel Puschmann, Frieder Ganz, and Payam M. Barnaghi. A knowledge-based approach for real-time iot data stream annotation and processing. In *GreenCom*, 2014.

[14] Alessandro Margara, Jacopo Urbani, Frank van Harmelen, and Henri E. Bal. Streaming the web: Reasoning over dynamic data. *J. Web Semant.*, 2014.

[15] Paula Reyero-Lobo, Enrico Daga, Harith Alani, and Miriam Fernandez. Semantic web technologies and bias in artificial intelligence: A systematic literature review.

[16] Ryan Shaw, Raphaël Troncy, and Lynda Hardman. LODE: linking open descriptions of events. In *ASWC*, 2009.

[17] Ioan Szilagyi and Patrice Wira. Ontologies and semantic web for the internet of things - a survey. In *IECON 2016*.

[18] Kerry Taylor, Armin Haller, Maxime Lefrançois, Simon J. D. Cox, Krzysztof Janowicz, Raul Garcia-Castro, Danh Le Phuoc, Joshua Lieberman, Rob Atkinson, and Claus Stadler. The semantic sensor network ontology, revamped. In *ISWC*, 2019.

[19] Riccardo Tommasini, Pieter Bonte, Emanuele Della Valle, Erik Mannens, Filip De Turck, and Femke Ongenae. Towards ontology-based event processing. In *OWLED*. Springer, 2016.

[20] Riccardo Tommasini, Pieter Bonte, Femke Ongenae, and Emanuele Della Valle. Rsp4j: An api for rdf stream processing. In *European Semantic Web Conference*, pages 565–581. Springer, 2021.

[21] Riccardo Tommasini, Mohamed Ragab, Alessandro Falcetta, Emanuele Della Valle, and Sherif Sakr. A first step towards a streaming linked data life-cycle. In *ISWC*, 2020.

[22] Riccardo Tommasini, Yehia Abo Sedira, Daniele Dell'Aglio, Marco Balduini, Muhammad Intizar Ali, Danh Le Phuoc, Emanuele Della Valle, and Jean-Paul Calbimonte. Vocals: Vocabulary and catalog of linked streams. In *ISWC*, 2018.

[23] Peter A. Tucker, David Maier, Tim Sheard, and Leonidas Fegaras. Exploiting punctuation semantics in continuous data streams. *TKDE*, 15(3):555–568, 2003.

[24] Dwight Van Lancker and et al. Publishing base registries as linked data event streams. In *ICWE*, 2021.

[25] Utz Westermann and Ramesh Jain. Toward a common event model for multimedia applications. *IEEE multimedia*, 14(1):19–29, 2007.

[26] 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. The fair guiding principles for scientific data management and stewardship. *Scientific data*, 3(1):1–9, 2016.