A Holistic View over Ontologies for Streaming Linked Data

Pieter Bonte [a,b,*], Femke Ongenae [b] and Riccardo Tommasini [c]
[a] *Department of Computer Science, KU Leuven Campus Kulak, Belgium,*
E-mail: pieter.bonte@kuleuven.be
[b] *Ghent University - imec, Ghent, Belgium*
E-mail: femke.ongenae@ugent.be
[c] *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 RSP 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

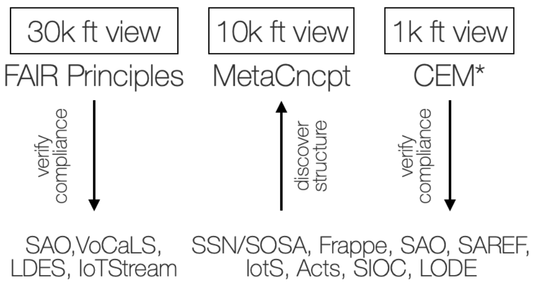

Fig. 1. The paper's contributions. A Three-folded perspective on the Knowledge Representation efforts for RDF Stream Processing respectively based on the FAIR Principles, a Meta [C]o[NC]e[PT]ualization, and the [C]ommon [E]vent [M]odel.

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) [69] 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 [32]. RDF

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

Stream Processing (RSP) is a subarea of SR that focuses on the processing of RDF Streams [64]. 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 [48] that, in turn, spawn research on Streaming Linked Data (SLD) [66, 71].

While data streams become more available on the Web, the community started discussing best practices to publish data streams in an interoperable manner. To 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)* [66]?.

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 [48]. Even though a number of domain-specific ontologies have been used in SLD applications, little has been done regarding the data modeling and knowledge representation efforts that SLD applications entail.

In this paper, we dig deeper into this claim by surveying the related literature and isolating such efforts. In particular, we investigated research papers that apply RSP, i.e. a subset of SR, as a solution. Like in similar works, we systematically select the papers, defining inclusion criteria and filtering methods. We extracted the used ontologies from these selected papers to model the data streams. We study such ontologies from three perspectives: (i) A **Thirty-Thousand Foot View**, which observes streams as Web resources analogous to dataset yet characterized by the velocity of changes; such view surveys existing practices for data modeling and KR for data streams. This view follows a top-down approach and starts from the FAIR principles [74] and verifies the compliance of several ontologies under survey. (ii) A **Ten-Thousand Foot View**, which gets closer to the streams and investigates its content; the result is a meta-conceptualization that empirically describes the structure of SLD vocabularies and ontologies. The definition of such a framework is guided by a review of existing stream processing conceptualizations [1, 4, 22]. (iii) A **Thousand Foot View** that narrows down even more until observing the internals of the items that populate a data stream, i.e., events. Thus, such a view leverages the Common Event Model [73] to study and explain how structurally SLD are presented. Our analysis shows how such a view complies with the inner parts of the stream representation.

Figure 1 summarizes our three-folded perspective, designed to highlight different aspects concerning knowledge representation for SLD by progressively zooming in. Indeed, higher levels offer a broader analysis than the ones below, encouraging a holistic view of the central concepts, i.e., Data Streams and their interrelations (30k), the classes and properties characterizing the content of data streams (10k), and the structure of the event as the unit of information that populate the streams (1k).

Outline: Section 2 introduces the necessary background to understand the paper's content. 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

This section presents the fundamental notions needed to understand the paper's content. In particular, we offer the survey methodology and the Streaming Linked Data lifecycle.

2.1. Survey Methodology

Our survey follows the guidelines of the systematic mapping research method [23], which has already been used successfully for surveys in the Semantic Web [55]. 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 SLD. Indeed, under the SR initiative several engines, query languages, and benchmarks were proposed to address SLD 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
– Linked Stream 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 [55], we apply *Metadata-based* filtering to the papers, screening their title, abstract, and publication venue and, then, we apply the *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 aims at expanding the relevant papers based on investigating their citations and related work. Especially for papers proposing SLD engines, it was very beneficial to investigate their citations as it revealed many use case papers.

Our analysis identified 32 papers from which we extracted 10 ontologies. The extracted ontologies are commonly used in one or more of the identified papers. The last step of our analysis was dividing the ontologies into two groups. The first group addresses SLD from a publication/discovery standpoint. Given the abstract view, we name such group Thirty-Thousand Foot View. The second group looks at SLD from a processing standpoint, which is a lower level of abstraction. Therefore, we name this group the Ten-Thousand-Feet View. We also notice that within the latter group, there is an even lower abstraction point of view which we name the Thousand Foot View and concerns the representation of data points within the streams. Figure 2 visualizes the selection process, while Table 1 lists the selected ontologies, their prefixes, each view they cover, and the papers they originated from.

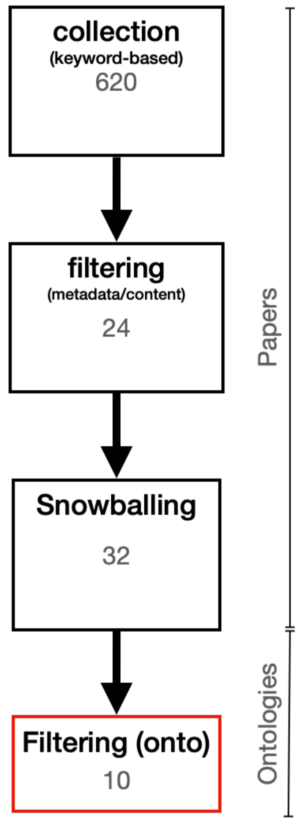

Fig. 2. Collection and Filtering methodology visualized.

2.2. Time(liness) and Events

In this section, we present some essential concepts that will recur alongside the remainder of the paper.

Time has always been under the scope of research in knowledge representation. Despite the number of proposals, there is still little agreement across communities, given the cascading consequences of temporal modeling. Directly related to the notion of time is the concept of change. Indeed, datasets are always subject to updates, ontologies are amended and revised, and sometimes, the answer to a given question changes too. Indeed, *variability* is an essential property of many concepts and, thus, represents a concern for knowledge representation and reasoning. Either way, temporality is represented with an (partially) ordered, discrete, and monotonic domain, e.g., natural numbers. Partial order allows the representation of simultaneous data items by assigning the same integer, a.k.a. the same timestamp. Discreteness and monotonicity are leveraged by the operator semantics to cope with the unbounded nature of input streams [68].

This paper also focuses on research works that leverage time as a measure of **timelines**, i.e., the need for processing data as soon as they are produced and before they are no longer useful. Later in section 3.1, we discuss foundational ontologies that are often imported to represent such concepts, below we provide a brief overview of the necessary notions.

Such works focus on abstractions such as streams or events. The former represents unbounded yet ordered data using temporal ordering, which is non-strict and is leveraged to define the processing semantics. In these regards, we say time plays the role of *punctuation*, i.e., it is used in stream processing systems to manage and control the flow of data and handle time-related tasks.

The latter, i.e., **events** are *occurrent*, i.e., they refer to the most general type of thing that happens in time (occurrence). Events are leveraged to describe the presence of change in a time-varying domain where facts are discovered/forgotten while time progresses. In this paper, we focus on works that operate using *instantaneous* events, which have an associated timestamp. Although interval-based time semantics is also possible [5], it is often limited at ontological level or represented by means of a duration statement.

Last but not least, it is worth mentioning *endurants* (aka *continuants*) that oppose to *occurrent* as they refer to things that happen through time (endurance), and whose identity is not implied by the time domain itself. In this paper, we focus on endurants in the context of query answering. Indeed, continuous queries are a family of queries in SLD that consume and produce streams and, thus, their evaluation is endless unless explicitly terminated.

2.3. Streaming Linked Data

RDF Stream Processing. Over the last decade, the Semantic Web community has made various proposals for languages to query RDF data in real time. The majority of these proposals involved extending RDF by adding timestamps or time intervals to each triple or graph. Notable languages in this category include C-SPARQL [14], Streaming SPARQL [75], CQELS-QL [52], and even more [31]. These languages expanded upon the SPARQL syntax to incorporate variations of sliding windows and, in some cases, introduced additional query functions. However, the semantics governing the behavior of these windows were not consistent, leading to varying operational behaviors. Consequently, these languages exhibited different syntax, semantics, and disagreements over the correctness of query results [30].

To address this issue, a unified formalization of continuous query processing over RDF streams was introduced in [30], known as RSP-QL, and a library RSP4J [64]. The former successfully integrates continuous query over RDF streams evaluation semantics and operational semantics of windows, enabling the characterization of existing SPARQL extensions for continuous querying. The latter aims at unifying existing RSP systems via a unique API inspired by RSP-QL primitives. Together, they contributed to pushing the state-of-the-art via the formalisation and prototyping of new languages [61] and systems [57]

Lifecycle. The Streaming Linked Data Lifecycle [18, 63] proposes several guidelines for managing data streams on the Web. Figure 3 depicts the whole lifecycle and highlights the *Model* and *Describe* steps, which both require a knowledge representation effort. The *Model* step takes care of modeling the content of the stream using a specific ontology-based knowledge representation. In contrast, the *Describe* step focuses on describing the stream itself as a Web resource. The latter aligns with the Thirty-Thousand Foot View, while the former aligns with the Ten-thousand and Thousand Foot View. Each of these steps requires 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 (2) is about structuring of stream data events; Step (3) focuses on converting streaming data into a machine-readable format; Step (5) is about serving data using protocols that enable continuous data access (e.g., WebSockets), and Step (6) relates to Web Stream Processing.

3. Selected Works

This section details the selected SR ontologies that will be investigated using the proposed Thirty-Thousand, Ten-Thousand, or Thousand Foot View.

3.1. Foundational Ontologies

We first describe four general ontologies that are frequently imported into the SR ontologies we will dis-

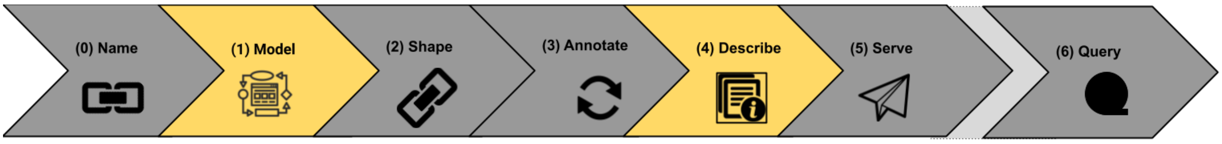

Fig. 3. Streaming Linked Data Life-Cycle from [18]

Ontology	Prefix	30kft	10kft	1kft	Projects
VoCALS	vocals	✓	✓$^-$		[26, 50, 66]
LDES	ldes	✓			[70, 72]
SSN/SOSA	ssn/sosa		✓	✓$^-$	[39, 50, 54] [3, 33, 46, 47] [25, 40–42, 45] [2, 56, 58]
SAREF	saref		✓	✓$^-$	[27–29]
IoT Stream	iots	✓	✓	✓$^-$	[3, 38]
SIOC	sioc		✓	✓	[8–10, 44]
LODE	lode		✓	✓$^-$	[13, 49]
ActS	acts		✓	✓	[7, 11]
Frappe	frp		✓	✓	[9]
SAO/CES	sao/ces	✓	✓	✓	[39, 54]

Table 1

Ontologies for Streaming Linked Data: Summary. (✓: supported, ✓$^-$: partly supported)

cuss 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 [59].

3.2. SLD-Specific Ontologies

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 [67] that aims at fostering the interoperability between data streams and streaming services on the web [67]. 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 [67].

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 [43].

The **Complex Event Ontology (CES)**[5] extends OWL-S 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 [36].

Linked Data Event Stream (LDES)[6] defines a collection of immutable objects that evolves over time, describing both historical and real-time updates. *LDES* uses the TREE specification[7] for the model-

[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
[5] http://citypulse.insight-centre.org/ontology/ces/
[6] https://w3id.org/ldes/specification
[7] https://w3id.org/tree/specification

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 2
Summary of Foundational Ontologies

ing 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 [71].

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 [34].

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 [62].

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 Events (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* [59].

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 [12].

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 [24].

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 1. Figure 4 visualizes the dependencies between the various selected SLD ontologies and the imported concepts or complete ontologies that they share. Certain SLD ontologies do not import a whole ontology, but rather import a limited subset of concepts of a certain ontology, this is visualized with the full dependency arrow in Figure 4, while complete imports of ontologies are visualized with dashed arrows. Note that the figure only depicts overlapping imports, i.e. imported ontologies that at least two ontologies share. Ontologies imported by a single SLD ontology are not depicted in order to keep a visual overview.

[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/

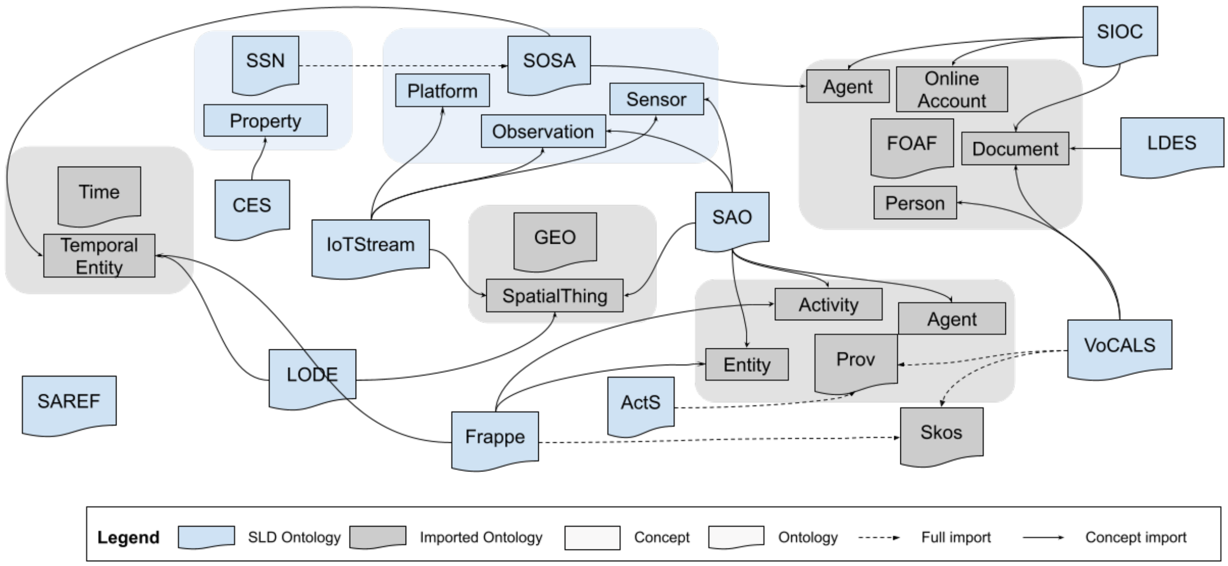

Fig. 4. Overview of depenencies between the selected SLD ontologies and the imported concepts/ontologies they share.

4. Thirty-Thousand Foot View: Web Streams

The thirty-Thousand-Foot View for SLD observes data streams as Web resources, i.e., the fundamental building blocks of the World Wide Web, and focuses on their metadata, governance, and provenance. Therefore, we reformulate our research question as follows:

RQ^{30K} *What characterizes the knowledge representation efforts for managing streaming (or highly dynamic) heterogeneous data, when the modeling focuses on streams and their content as referentiable Web resources*

Only four of the ten selected ontologies have the notion 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

Our analysis builds upon the preliminary adaptation of the FAIR principles proposed in [66]. The original FAIR Principles [74] are reported 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.

Notably, the Thirty-Thousand Foot View does not aim at assessing whether existing ontologies follow the FAIR principles themselves (as similar effort has been done in previous research [53]). Instead, the analysis investigates if existing ontologies allow to share FAIR streaming data on the Web. The analysis focuses on the ontological level and its (potential) applications. Definition 1 introduces the notion of Web Stream, which is a prerequisite for identifying streams on the Web.

Definition 1. *A Web Stream is an unbounded ordered collection of pairs (o, i), where o is a Web resource, and i is event-wide metadata selected to establish a form of punctuation such as a timestamp.*

Definition 1 captures the double nature of Web Streams, which are both a resource (indeed they are identifiable) but also "contain", i.e., refer to other resources on the Web. Such a two-fold nature extends to the data and metadata levels. Therefore, we can distinguish between stream-wide and event-wide (meta)data, which relate to the stream resource and its content, respectively [65]. Stream-wide (meta)data contains information about the whole stream, for instance, who is the publisher, or a list of known consumers; on the metadata level, we find the date when the stream was first issued, descriptive statistics about the data or the formats in which the stream is available. Event-wide (meta)data concern each Web resource within the stream. For instance, a resource can refer to a domain-specific entity, which in turn depends on where the stream is originally from (e.g., for an IoT stream monitoring the location of people, an entity can be a given Point of Interest or a person). The role of Event-wide metadata relates to the event order, duration, or location. Notably, a punctuation mechanism that is needed to enable continuous processing is usually based on time. However, it can be generalised to any Boolean predicate related to order that leverages event-wide metadata [68].

4.2. Discussion

We now analyze the selected ontologies, w.r.t. the FAIR data principles. While Table 3 summarise the answers to the individual principles, we organize the discussion along the following dimensions by answering the related questions:

D.1 **Identity (F1, F3, A2)**: *Is it possible to use IRIs or DOIs to identify the Web Stream and/or the referred resources in ontology X?*

VoCaS, *LDES*, and *IoTStream*, introduce very similar concepts that lead to instantiating referencable Web Streams. More specifically, *VoCaLS* includes the notion of `voc:Stream` specifically to represent an unbounded dataset on the Web; *LDES* introduces the notion of `ldes:EventStream` as an append-only collection of immutable elements, and assigns to it a retention policy; Elsaleh et al. include in their IoT Stream ontology the notion of `iot:IoTStream`. *SAO* goes one step further, allowing its users to identify the resources within the stream as `sao:StreamData` or `sao:StreamEvent`; the two classes distinguish the raw elements from those produced by some analysis. The class `sioc:Thread` and the more generic `sioc:Container` refer to a collection of elements. However, they do not explicitly mention an ordering relation between them. Similarly, *ActS* includes the concept of `OrderedCollection` that aligns with the Web Stream Conceptualisation, while individual activities represent elements in the collection. Finally, *LODE* allows only the instantiation of individual events without conceptualizing the Web Stream. Although the presence of a class that aligns with the conceptualization in Definition 1 does not prevent instantiating the stream anonymously (with blank nodes), it allows the FAIR usage with transparent IRIs/DOIs (F1).

D.2 **(Meta)Data Semantics (F2, I1, R1)**: *Can the ontology X capture the (meta)data semantics at stream and event level? What formalism was used for the modeling efforts?*

Among the selected ontologies, only five have a conceptualization that can be coherently aligned with Web Streams and, thus, allow representing stream-level data. *VoCaLS* and *LDES* allow specializing RDF Streams, but they do not specify anything regarding the event-level semantics. On the other hand, *SAO/CES*, *IoTStream*, *SAREF*, and *SSN/SOSA* focus only on representing data only at the event level, following a commonly accepted ontology design pattern for modeling sensor measurement in RDF based on observations. Also *LODE*, and *Frappe* neglect the stream level (as seen before) and focus only on the event-level dimension for data and metadata. Finally, *SIOC*, *ActS* are the only two ontologies that can possibly define data at both stream and event level, nonetheless, with some limitations wrt. the conceptualisation of Definition 1.

Regarding metadata, *VoCALS* supports to descriptive information about the resources, e.g., name and owner, and contextual information, e.g., the vocabulary used to annotate the stream content, as well as stress on the specification of a license (R1). Instead, *LDES* explicitly supports only contextual metadata as it relies on the TREE specification, which also includes a license (R1). Notably, also *SAO/CEO* supports licensing via the imported ontology *QOI*. Although not explicitly declared, the same approach would be possible in *SIOC* and *ActS*, as both have a concept that can be aligned to Web Streams. Finally, neither *SIOC* and *ActS*, nor *SAO/CES*, *IoTStream*, *SAREF*, and *SSN/SOSA* do explicitly define event level metadata.

Finally, all the selected ontologies use OWL (Frappe, VoCals, SAREF, SAO/CES, IoTStream, SSN/SOSA) or RDFS (Activity Streams, SIOC) as ontological languages to implement their formalization.

FAIR	Dimension	VoCaLS	SAO/CES	LDES	IoTStream	SAREF	SIOC	LODE	ActS	Frappe	SSN/SOSA
F1	Identity (S)	✓		✓	✓		✓^{U}		✓		
	Identity (E)		✓		✓	✓	✓	✓	✓	✓	✓
F2	Quality (G)		✓		✓						✓
	Quality (D)	✓	✓		✓		✓		✓		✓
	Quality (C)	✓	✓	✓	✓		✓		✓		✓
	Semantics (S)	✓		✓	✓		✓		✓		
	Semantics (E)	✓			✓	✓	✓	✓	✓	✓	
F3	Identity	✓		✓	✓		◇^{U}		◇		
	Data Model	✓^{S}	✓	✓^{S}	✓	✓	✓	✓	✓	✓	
F4	Quality (S-I)	✓		✓	✓		✓		✓		
	Quality (E-I)		✓			✓	✓	✓	✓	✓	
A1	Protocols	✓	✓	≈	✓		≈			✓	
A2	Identity	✓		✓	✓		◇		◇		
	Protocols	✓		✓	✓		✓		✓		
I1	Semantics (S)	✓		✓	✓		✓		✓		
	Semantics (E)		✓			✓	✓	✓	✓	✓	✓
I2	Referencing	≈		✓				✓	✓		
I3	Referencing	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
R1	Semantics	✓	✓		✓						
R2	Quality (P)	✓	✓		✓	✓				✓	✓
R3	Data Model	✓	≈	✓	≈		≈		≈		

Table 3

Summary of the thirty-thousand-foot view, i.e., compliance of the Selected ontologies (top) with FAIR Principles (left) and our analysis dimensions (left) (Terminological Level Only) Legend: ◇=possible; ✓=supported; ≈=partially supported; [S]tream; [E]vent; [G]general; [D]escriptive; [C]ontext; [P]rovenance; [I]indexing; [U]nordered; [N]ot [A]pplicable.

D.3 **Data Models (F3, R3) and Adequate Protocols (A1, A2)**: *Can adequate access protocols for streaming (meta)data be defined using ontology X? Are the (meta)data appropriately licensed, and is the licensing specific to the stream? Can (meta) data stream be represented using the RDF data model in ontology X?*

All the selected ontologies support and encourage using RDF (Streams) to represent data and metadata (F3). However, not all focus on the stream and event levels. *VoCaLS* and *LDES* even explicitly include an RDF Stream specialization of the generic data stream. Although choosing an adequate protocol for sharing (meta)data on the Web usually means HTTP, it does not directly apply to streaming data. Regarding sharing, *VoCALS* and *LDES* adopt the convention, introduced initially by Barbieri et al. [15], who suggested sharing the stream metadata in a separate document accessible via HTTP while adopting a more suitable protocol for the stream content (F3, A2). Notably, the same approach would be possible with the *SIOC* and *ActS* given that we could find an alignment with the concept of a Web Stream. Finally, except *LDES*, which inherits the HTTP access assumption from *TREE*, the other ontologies include a specific abstraction that aims at generalizing access to the streaming data. Still, they do not recommend explicitly any protocols except IoTStream (e.g. RESTful, NGSI-9, MQTT, CoAP etc.), i.e., `voc:StreamEndpoint`, `sioc:Space` (is a place where data resides, e.g. on a website, desktop, fileshare, etc.) `iots:Service`, `saref:Service`, `ces:EventService`.

D.4 **Data Quality (F2, F4, R2)**: *What dimensions of data quality does ontology X consider?*

Among the selected ontologies, only *SSN*, *SAO/CES*, and *IotStream* explicitly focus on data quality by including specific classes and properties. Their modeling is thorough, and it includes all the traditional data quality dimensions like Accuracy, Volatility, and Completeness. For the sake of the analysis, we discuss them as part of a *General* definition [51], distinguish them from other aspects related to *Descriptive* and contextual metadata, or traceability, which is another essential dimension of data quality that is explicitly named by FAIR principles (R2) as Provenance.

SSN System Capabilities Module[12] includes several dimensions, e.g., `ssn-system:ResponseTime`, `ssn-system:Frequency`, or the conceptualisation of `ssn-system:Drift`. *SAO/CES* and *IotStream* import many dimensions from the Data Quality Ontology *QOI* [13], for example `qoi:Accuracy`, or `qoi:Completeness`, or `qoi:Jitter`.

Moreover, *VoCALS*, *LDES*, *IoTStream*, as well as *SIOC*, *SSN*, and *ActS* (although implicitly), includes classes and properties for describing the streams and linking to contextual resources, e.g., services that can contribute to the quantification of the quality level.

Regarding provenance (R2), all the ontologies, except for LDES, which is not focused on processing, include dedicated classes and properties for tracking the provenance of streaming analysis, i.e., `vocals:Task` and `vocals:Operator` for representing queries, `ces:StreamAnalysis` and `ces:EventPattern` for aggregations and complex event recognition, for spatio-temporal analyses `frappe:Synthetize` and `frappe:Capture`, and `saref:Function` and `iots:Analytics` or `ssn:Procedure` for continuous processing over the observation streams.

Finally, *LODE* does not support any data quality dimension. At the same time, all the ontologies that allow the usage of explicit identifiers support indexing and searching for URIs.

D.5 **FAIR Referencing (I2, I3)**: *Does ontology X provide explicit mechanisms for referencing external (FAIR) resources, such as connecting the stream and its items?*

Linking across resources is essential to the Semantic Web and, more generally, interoperability. Also, the FAIR principle encourages this, translating at the ontological level with the explicit possibility of linking to external resources (outside the (meta)data semantics). Not all the ontologies support it explicitly, but only *VoCALS* allows to connect a given Web Stream with vocabularies, mapping files, and/or ontologies; *LDES* via the `tree:member` inherited from TREE, which allows connecting any referentiable resources to the stream or its elements; *ActS*, with the class `Link` that is meant to be an indirect reference to another resource, and finally *LODE*, which includes two properties: `involved` and `involvedAgent`, that aimed at representing any physical, social, mental object or an agent involved in an event.

[12]https://www.w3.org/TR/vocab-ssn/#System-capabilities
[13]https://mobcom.ecs.hs-osnabrueck.de/cp_quality/.

```
:CadornaTrafficStream a ssn:Output, vocals:Stream .
:TrafficFlowSensing a sosa:Procedure, sao:StreamEvent;
  prov:used :CadornaTrafficFlow ;
  ssn:hasOutput :CadornaTrafficStream.
:CadornaTrafficSensor a sosa:Sensor ;
  sosa:observes  :TrafficFlow  ;
  ssn:implements :TrafficFlowSensing .
:CadornaTrafficFlow a sosa:Result, sao:StreamData ;
  prov:wasDerivedFrom :CTObservation .
:CTObservation a sosa:Observation;
  vsd:TimeVaryingGraph, event:Event;
  ssn:observedProperty :TrafficFlow ;
  sosa:hasResult :CTSensorOutput ;
  event:time [a time:Instant ;
            time:inXSDDateTime "2023-01-01T00:00:00"^^
                xsd:dateTime ] .
```

Listing 1 Combination of VoCALS with SAO and SSN Ontologies to increase FAIR coverage. Prefixes omitted.

Unfortunately, there is no way to verify whether the linked resources follow the FAIR principles by only looking at the ontological level. However, if we only limit our indirect assessment to the selected ontologies, any interlinked Stream that reuses a combination of the selected one would be FAIR.

It is important to note that every ontology does not need to cover all aspects. It is possible to combine ontologies with different capabilities to obtain complete coverage. A combination of VoCALS with SAO and SSN was already explored in the original VoCaLS paper [67] and is reintroduced in Listing 1. We utilized the SOSA/SSN vocabularies to represent the source device and the observation data it produces, and SOA to describe information about the output of a stream observation, in addition to capturing the stream and streaming services metadata. The listing reflects an interpretation of Table 3, which shows that the combination of VoCaLS with complementary ontologies such as SOA or IoTStream can increase the FAIRness of the streams.

4.3. Best Practices

From our discussion emerges a clear need for greater emphasis on adhering to the FAIR principles and addressing the challenges specific to stream reasoning, ensuring that data streams are not only analyzed in real-time but are also readily discoverable, accessible, interoperable, and reusable for both current and future research and applications.

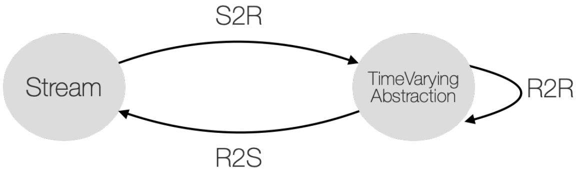

Fig. 5. Streaming Linked Data Abstractions

When modeling an ontology for SLD, the primary goal should be to maximize FAIR coverage. The rapid development of SLD technologies has led to overlook these aspects. Indeed, it's not uncommon for a single ontology in this domain to fall short of meeting all the FAIR principles comprehensively (see Table 3). In such cases, it's advisable to pursue a strategy of combining multiple ontologies to bridge these gaps and maximize FAIR coverage collectively, thereby enhancing the effectiveness of stream reasoning systems.

BP_1^{30k} Maximize FAIR coverage in new design;
BP_2^{30k} Combine ontologies to maximize FAIR coverage not just for domain modeling compliance;

5. Ten-Thousand Foot View: Streams' Structure

The Ten-thousand Foot View focuses on the ontological level and analyses the nature and nurture of the conceptualization of the selected ontologies used for representing streaming data within a given domain.

RQ^{10k} *What characterizes the knowledge representation efforts for managing streaming (or highly dynamic) heterogeneous data, when the modeling efforts are tailored for a given application domain and must consider domain-specific entities?*

According to our Thirty-Thousand Foot View analysis (see Table 3), only eight of the ten selected ontologies describe concepts to represent the streaming data at the event level. These eight ontologies include SSN/SOSA, SAREF, IoTStream, SIOC, LODE, ActS, Frappe, and SAO/CES. The other ontologies are not included in this discussion.

5.1. Analysis Framework

In the related literature [4, 31, 48], 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. [4] introduced such data dichotomy to the extent of formalizing relational Continuous Queries. Dell'Aglio et al. [30] extended it later on for RSP. In this work, we focus on SLD 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.*

SLD focuses on query answering over RDF Streams, i.e., Continuous Computations (see Definition 3) that 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.

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

On the other hand, Time-varying abstractions represent the result of Continuous Computations and, as the term suggests, capture the changes that occur to data as a function of time. Definition 4 formalizes the notion and specializes the definition.

Definition 4. *Time-varying Abstractions (TVA) are functions that map the temporal domain to finite entity sets that relate to 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.

Many extensions of SPARQL exist [32] to perform Continuous Queries over RDF Streams, and the RSP-QL [30] reference model aims at unifying the formal semantics of existing SPARQL extensions. Its abstraction can be found in Figure 5. 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 their original version that is applicable to static data (R2R). Finally, an operator's class that transform back Time-varying data into streams is called Relation-to-Stream (R2S). According to RSP-QL, a Time-varying RDF Graph results from applying 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 [48].

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 align with the meta-conceptualization described above. For this reason, LDES is not taken into account in this discussion.

An ontology used for SR typically consists of five levels, i.e., *L1* the *instantaneous level* identifies the part of the ontology 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 [48]. For the sake of completeness, we also include a *time-agnostic level L3*, which identifies those ground terms independent of time. *L4* the Time-varying level includes entities whose state evolves. 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, we leave a deeper investigation of L5 as future work due to the lack of space.

The detailed analysis of the selected ontologies is presented below and summarized in Table 4.

The decision diagram in Figure 6 is structured to guide knowledge workers operating within the SLD context at the Ten-Thousand Foot View. The diagram helps determining the classification of ontology concepts based on time. For instance, if one is determining if "time is part of the conceptualization," and the answer is "no," then the concept is "Time Agnostic." If the answer is "yes," further decisions based on "occurrence", "endurance," and "change" lead to the classification of the concept into one of the other levels. The diagram provides a structured approach to categorizing ontology concepts by their relationship with time, which aligns with Definitions 2,3,4, and the general notion of time presented in Section 2.

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 present 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 occurs. 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 online communities' population and logical location. LODE does not include concepts at L2. VoCaLS includes `Stream` and `RDFStream` as static concepts. They are meant to represent streams as resources (to be continuously consumed).

Time Agnostic (L3). Neither Frappe nor SAO/CEO, initially designed for SR/RSP applications, directly include L3 concepts. On the other hand, 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.

Ontology	Instantaneous (L1)	Static (L2)	Time Agnostic (L3)	Time-varying (L4)	Continuous (L5)
SSN/SOSA	Observation, Result	Sensor, Platform,	ObservableProp., Measure	Actuation, Result	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.

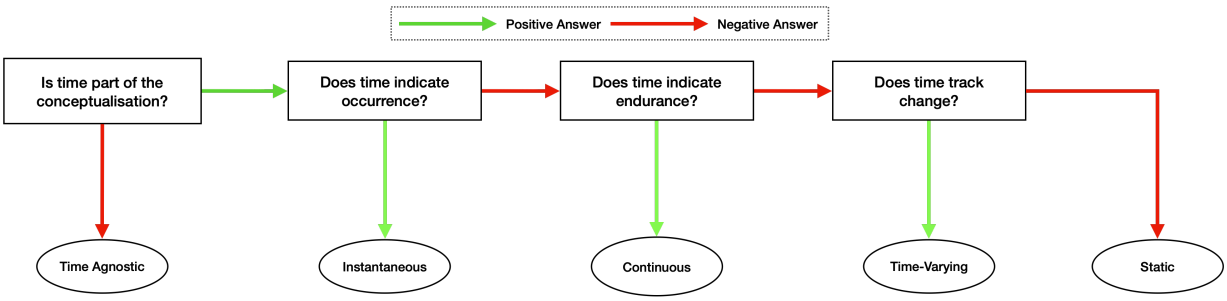

Fig. 6. Decision Diagram for assigning the meta-structure in the Ten-Thousand Foot View. Red Arrow is "no", Green Arros is "yes".

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 [30], i.e., `TimeVaryingGraph` that represents the Time-varying equivalent of an RDF Graph, and `SDS`, which is a collection of TimeVaryingGraphs. Moreover, VoCaLs explicitly mentions continuous 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 their complexity across different temporal levels, facilitating the alignment with SR applications.

5.3. *Reasoning Capabilities*

The selected ontologies include complex concepts requiring definition consisting of expressive language constructs. Such constructs have, in turn, an impact on the expressivity of the including ontology. In the following, we discuss these nuances focusing on how they related to our meta-structure (see Figure 6). Moreover, we discuss opportunities for reasoning optimizations. Table 5 summarises the expressivity of each ontology in terms of minimum OWL2 Profile and De-

Ontology	OWL2 Profile	Description Logic
SOSA	OWL2 RL, QL	ALI(D)
SSN	OWL2 DL	ALRIN(D)
SAREF	OWL2 DL	ALCIQ(D)
IoT Stream	OWL2 DL	ALCHI(D)
SIOC	OWL2 DL	SHI(D)
LODE	OWL2 DL	ALHF
ActS	OWL2 DL	ALCHN(D)
Frappe	OWL2 DL	SROIN(D)
$\text{Frappe}_{noimports}$	OWL2 QL	ALI(D)
SAO	OWL2 RL	ALH(D)
CES	OWL2 RL	ALH(D)
VoCALS	OWL2 DL	SRIN(D)
$\text{VoCALS}_{noimports}$	OWL2 EL, QL, RL	ALH

Table 5

Ontology expressivity in terms of OWL2 Profile and Description Logic

scription Logic (DL)[14]. Notably, most ontologies requires very expressive languages, i.e. OWL2 DL Profile, to be fully interpreted. The mismatch between the high complexity of the reasoning algorithms required to interpret these ontologies and the frequency at which data is updated in SR applications [21], makes these ontologies ill-suited for SR applications at first glance. For the ontologies with import statements, i.e., Frappe and VoCALS, we distinguish between the core ontology's expressivity with and without its imported ontologies. We can see that both ontologies owe their high expressivity to their imported ontologies, as their concept definitions are much lower in expressivity.

We now zoom deeper into various complex definitions and their structural relation to SR tasks. As the goal in SR applications is to reason upon the events in the stream and combine them with other contextual data, we investigate complex concept definitions that span across levels (L1-L5), stressing in particular on L1. We define complex concept definition in DL notation, i.e. $B \sqsubseteq H$, which informally could be interpreted as '*if B then H*'. In turn, B and H can be complex definitions constructed from conjunctions ($\sqcap$), disjunctions ($\sqcup$), existential ($\exists$), or universal ($\forall$) quantifiers.

We focus on reasoning on instance level (ABox), through definitions defined across the five ontology meta-structures. We differentiate between complex definitions using either existential in the subclass definition (i.e. B) or universal quantification in the superclass definition (i.e. H). For example, $\exists observes.Temperature \sqsubseteq TemperatureSensor$ describes a existential value restriction, i.e., an individual that `observes` the property `Temperature` can be *inferred* as a *TemperatureSensor*; while $Observation \sqsubseteq \forall madeBySensor.Sensor$ describes a universal value restriction, i.e., any individual that has assigned the `Observation` class can only be made by a `Sensor`, and otherwise the ABox would result inconsistent.

We identified four interesting reasoning perspectives based on the position of L1 in the complex definitions, i.e. either in B or H. With *Other* we denote all other levels, except L1. Table 6 summarizes the identified reasoning perspectives for each ontology.

Perspective 1 ($L1 \in B, Other \in H$): concepts of L1 are present in B, while H contains concepts outside of L1. This means that the event in the stream needs to be *enriched* with data outside of L1.

- *Existential:* This kind of definition implies that the events in the stream influence the classification of the data defined outside of L1. None of the ontologies have predefined definitions in this perspective, except for object property domain and range definitions. For example, SAREF defines `Device` (L2) as the domain of the property (makesMeasurement), which has `Measurement` (L1) as a range ($\exists makesMeasurement.T \sqsubseteq Device$). We typically find definitions of this kind in application-specific ontologies. For example, in [27], the authors extend SSN with `FaultyTemperatureSensor` (L2), which is a `Sensor` (L2) that made an `Observation` (L1) which has a certain `Symptom` that is a `TemperatureValueDeviation`[15] ($Sensor \sqcap \exists madeObservation.(Observation \sqcap \exists hasSymptom.TemperatureValue-Deviation) \sqsubseteq FaultyTemperatureSensor$).
- *Universal:* many ontologies use universal quantification to define restrictions that span L1 into either L2 or L3. For example, SSN restricts an `Observation` (L1) as something that can only be made by a `Sensor` (L2). ($Observation \sqsubseteq \forall madeBySensor.Sensor$)
- *Efficiency:* Reasoning about the existential definitions in this perspective is non-trivial as the reasoning task requires reclassifying the more static data based on the content of the stream. Reasoning on the universal restrictions is more efficient as it can be optimised by materializing the more static data, such that the restrictions on the events in the streams can

[14] We refer the reader to Baader et al. [6] for a complete introduction to DLs, as it is out-of-scope for this paper.

[15] Both *Symptom* and `Temperature ValueDeviation` are application specific and not part of the SSN ontology.

be computed by linking the event to the materialized static data and computing the consistency only of the instances defined in the event itself. This is similar to the idea of SubSet Reasoning [17] where a subset of the materialized data is extracted to reason upon the data in the stream.

Perspective 2 ($L1 \in H, Other \in B$): concepts of L1 are defined in H, while B contains concepts outside of L1. This also means that the event in the stream needs to be *enriched* with data outside of L1.

- *Existential:* None of the ontologies have definitions in this perspective, except for object property domain and range definitions. For example, SAREF defines `Measurement` (L1) as the domain of the property `measurementMadeBy`, which has `Device` (L2) as a range. However, we see that most of this perspective is defined directly in the application logic that builds on these ontologies. For example, the CityPulse project [54] defines ASP rules in this perspective, while [21] defines a `CO2Observation` as an `Observation` (L1) that is observed By a `Sensor` (L2) that observes the `Property` (L3) CO2. ($Observation \sqcap \exists madeBy.\exists observes.CO2 \sqsubseteq CO2Observation$)
- *Universal:* Mostly the IoT ontologies use universal quantifications to define restrictions in this perspective. For example, SSN defines that a `Sensor` (L2) can only make observations of the type `Observation` (L1) ($Sensor \sqsubseteq \forall madeObservation.Observation$).
- *Efficiency:* the existential quantifiers in this perspective allow to materialize the more static data and perform the reasoning on a restricted set of data around what is defined in the event [17] or try to cache the reasoning steps that are needed to reasoning on the event data [21].

Perspective 3 ($Other \notin H, Other \notin B$): This perspective of definitions is defined solely on L1, allowing reasoning to be performed without any enrichment of the more static data in the other levels.

- *Existential:* None of the ontologies have definitions with existential quantifiers in this perspective, however, as an example, we could imagine an application extension of SIOC that defines `AcademicPosts` as `Posts` (L1) that describes a certain topic as the literal `"academic"`.

Ontology	**Reasoning Perspective**			
	1	2	3	4
SOSA	-	-	-	-
SSN	U	U	U	U
SAUEF	U, E_D	U, E_D	U	U, E_D
EoT Stream	U, E_D	U, E_D	-	U, E_D
SEOC	E_D	E_D	-	E_D
LODE	-	-	-	-
ActS	E_D	E_D	U	U
Frappe	E_D	E_D	-	-
SAO/CES	U, E_D	E_D	-	E_D
VoCALS	-	-	-	E_D

Table 6

Various reasoning classes that influence an ontologies SR abilities. (U = Universal, E = Existential, E_D = Domain/range Existential

- *Universal:* Most of the IoT ontologies have again definitions in this perspective, e.g. SSN defines a `Observation` (L1) as something that only has instances of the type `Results` (L1) as result ($Observation \sqsubseteq \forall hasResult.Result$).
- *Efficiency:* This perspective is efficient in terms of reasoning as it does not require any interaction with the more static data defined outside of L1.

Perspective 4 ($L1 \notin H, L1 \notin B$): This perspective of definition are all defined outside of L1. Allowing the reasoning the be done independent of the content of the stream.

- *Existential:* Again none of the ontologies have predefined definitions in this perspective. However, we can again find examples in the application logic of certain projects. [28] defines a `TemperatureSensor` (L2) as a `Sensor` (L2) that observes the Property `Temperature` (L3) ($Sensor \sqcup \exists observes.Temperature \sqsubseteq TemperatureSensor$).
- *Universal:* Similar to **Perspective 3**, many of the IoT ontologies use universal quantifiers to define restrictions for this perspective. For example, SSN defines a `Sensor` (L2) as something that can only observe `Observable- Properies` (L3) ($Sensor \sqsubseteq \forall observes.ObservableProperty$).
- *Efficiency:* This perspective can be precomputed as reasoning can happen independent of the events in the stream.

So even though most ontologies were very expressive at first glance, they mainly use this expressivity to define restrictions on the various concepts, while the inference tasks are typically reserved for application specific logic.

5.4. Best Practice

At this level of analysis, we recommend to follow four valuable lessons to enhance the effectiveness of data processing. Firstly, practitioners shall carefully examine the expressivity of imported ontologies and striving to limit their complexity, ensuring that the ontologies utilized align closely with the specific requirements of their applications. Indeed, we observed that despite the attempt of keeping the ontology profile down to OWL 2 QL, resolving all the imports causes the overall profile to be much more complex (OWL 2 DL). Secondly, it is advisable to maintain a low reasoning expressivity when defining the concepts related to events. Recent results on hierarchical reasoning show how SLD applications could benefit by limiting to such modeling practice [20], which also helps streamline the processing of streaming data by avoiding unnecessary complexity in stream reasoning tasks. Furthermore, it's essential to avoid Reasoning Perspective 1, where event data significantly influence the classification of more static data. This approach can be challenging to optimize and may lead to inefficiencies in data handling [17]. When selecting ontologies for integration in the stream reasoning context, aim for those that exhibit clear differentiation in their meta-structure (see Figure 6), as identifying the change frequency of instances based on their assigned concepts allows to optimize the processing. Indeed, differentiation allows to avoid redundancy and promote effective knowledge representation and data integration within this dynamic and evolving domain [40].

By heeding these lessons, the field of SLD can better manage the intricacies that occur when modeling a domain that presents streaming data and continuous information needs.

BP_3^{10k} Check the expressivity of the imported ontologies and try to limit the imported expressivity.

BP_4^{10k} Keep the reasoning expressivity of the concepts that define the event as low as possible.

BP_5^{10k} Avoid Reasoning Perspective 1 in which the event data influence the classification of the more static data, as it is not trivial to optimize.

BP_6^{10k} Aim for a clear differentiation in the ontology meta-structure.

6. Thousand Foot View: Streams' Content

The Thousand Foot View of SLD 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.

RQ^{1k} *What characterizes the knowledge representation efforts for managing streaming heterogeneous data when the modeling efforts are limited to the event level?*

6.1. Analysis Framework

The Common Event Model (CEM) was initially proposed by Westermann and Jain for multimedia applications [73]. 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 [68]. In SR/RSP, punctuation relates to the stream shapes, e.g., Graph, Triple, Predicate, as well as with the notion of Event Types [32]. At the ontological level, this reflects on the levels of conceptualization, especially L1. Thus, we introduce the following notion:

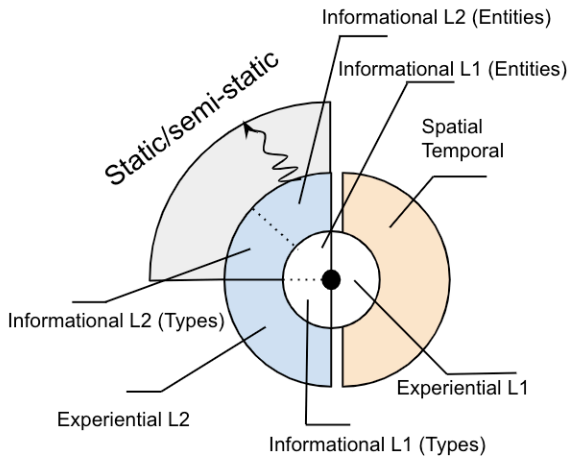

Fig. 7. Kernel Structure.

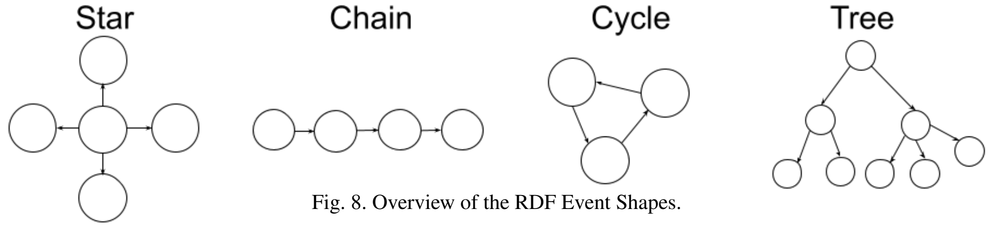

Fig. 8. Overview of the RDF Event Shapes.

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

Our analysis highlights the relation between the Kernel and the meta-conceptualisation levels (cf. Section 5). Figure 7 depicts such relation enumerating the levels across the CEM dimensions, which are:

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, sensor measurements, or audio 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 8. Note that ontologies allow to model events using multiple shapes.

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 [63].

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.

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 7 and for the Spatial and Temporal discussion in Table 8.

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 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

	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 7

Overview of Ontology Kernel analysis for Informational and Experiential information.

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 [19]. 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 *Sensor*s) 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 change 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` 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-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 a 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`.

Structural. Figure 9 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 the 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

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 8

Overview of Ontology Kernel analysis for Spatial and Temporal information.

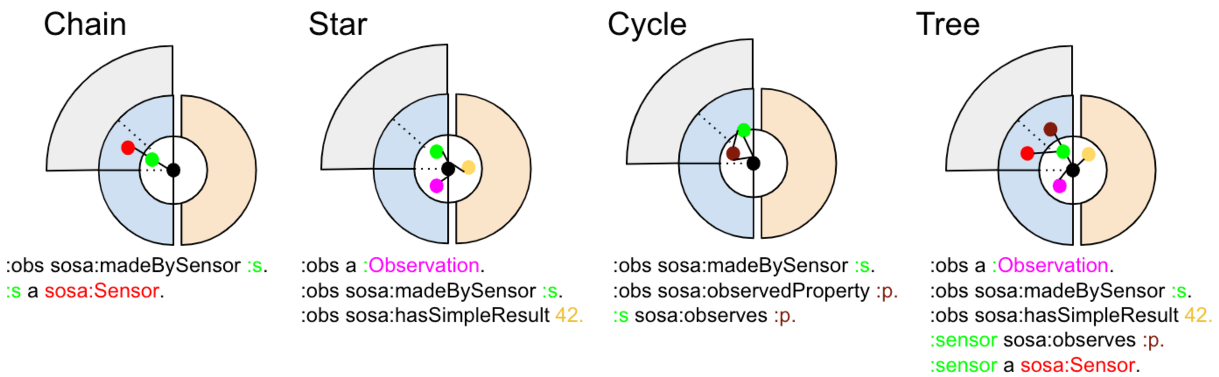

Fig. 9. Mapping of the RDF structures on the Event Kernel using the SOSA ontology.

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

Table 9

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 10

RDF shapes alignment with the kernel and ontology levels.

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 9 and 10 summarize the analysis.
Understanding the structure of the events is important as it opens many opportunities for optimizations, as it allows to clarify how a query can optimally interact with the events. For example, Stars could be represented as a table (instead of an RDF graph) allowing part of the querying to be offloaded to lower-level processing techniques that operate before the conversion to RDF which can improve performance [11]. Fernandez et al. [35] showed that identifying regularities in the structure of the data in the stream allows to improve transmission by structure-tailored compression techniques. Furthermore, Bonte et al. [16] showed that understanding the structure of the events in the stream allows to optimize the continuous query evaluation process. These kinds of optimizations then on their own can lead to better modeling guidelines for SLD ontologies.

Composition. Most ontologies allow some sort of composition through logical reasoning between the kernel and data that is modeled outside of the kernel, as discussed in Section 5.3. However, it is worth noting that some ontologies allow to define compositions that go beyond traditional logical reasoning. SOA/CES allows to define temporal patterns through the Complex Event Processing (CEP) definitions supported by the CES ontology. These CEP definitions allow defining the composition of various events that have a temporal dependency. Frappe allows compositions by defining aggregations on the captured data through statistical inference. Similarly, IoTStream allows to define how different *Analytics* have been computed on the data stream that also allows some sort of statistical inference to perform composition over various events. SAO has similar functionality through its *StreamAnalysis* concept, and even predefines a number of analyses, among others *KMeans*, *MovingAverage* and *DiscreteCosineTransform*.

6.3. Best Practices

Finally, at the lowest level of our analysis, we share several key lessons that have emerged. To promote streamlined processing in real-time environments, it is advised to keep the core kernel of the data model as concise as possible or at least limit the expressiveness of the ontological fragment that it uses. Indeed, the more properties constitute the kernel, the higher the risk for encountering unexpected dependencies with static knowledge (see Perspectives in Section 5.3). Additionally, the adoption of event structures that can be easily translated into simpler representations, such as the Star model, can be optimised for matching independently from the window [52]. When incorporating temporal information, adhering to widely accepted temporal concepts like *time:TemporalEntity* fosters uniformity and bolsters interoperability. Likewise, for spatial information, the reuse of established concepts from ontologies like "geo" or "geosparql" is favored over introducing custom location-specific terms, contributing to more standardized and compatible data representations. Indeed, we notice high diversity across the adopted spatio-temporal concepts. However, having a shared and agreed-upon conceptualisation of space and time is an essential aspect of SLD applications.

These lessons collectively advance the field of SLD, enabling more effective management and utilization of dynamic and evolving datasets.

- BP_7^{1k} Keep the kernel as small as possible.
- BP_8^{1k} Rely on an event structure that can easily be translated to simpler representations, such as the Star.
- BP_9^{1k} When modeling temporal information, regardless of the need for point or time semantics, use widely accepted existing temporal concepts such as *time:TemporalEntity* in order to pertain uniformity and improve interoperability.
- BP_{10}^{1k} For spatial information, refrain from introducing custom location-specific concepts and reuse concepts from the *geo* or *geosparql* ontologies.

7. Related Surveys

Dell'Aglio et al. [32] 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. [48] 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. [60] 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. [37] 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 SLD. 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 SLD 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 SLD 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). Furthermore, we aim to zoom in further on the processing part, i.e. L5 of the Ten-Thousand Foot View and the Causal dimension of the Thousand Foot View.

Our analysis introduced a number of reasoning perspectives, which opens opportunities to design an ontology profile that opens the possibilities for various reasoning optimization that can be identified by the different perspectives. Our analysis frameworks also open various directions in terms of optimized processing. For example, the Ten-Thousand-Foot View opens optimizations by explicitly defining the interaction between the data in the stream (instantaneous level) and more slowly changing data. Similarly, the Thousand Foot View opens optimizations by identifying the different shapes of events. In terms of knowledge representation, we have identified opportunities to define ontology metrics for SLD ontologies, starting from our analysis frameworks.

Most importantly, our analysis frameworks can aid to evaluate future ontologies for SLD and serve as a guideline for high-quality knowledge representation.

Acknowledgments:

This work was partly funded by Research Foundation Flanders (FWO) (1266521N). R. Tommasini is supported by the French Research Agency under grant agreement nr. ANR-22-CE23-0001 Polyflow.

References

[1] Tyler Akidau, Robert Bradshaw, Craig Chambers, Slava Chernyak, Rafael J Fernández-Moctezuma, Reuven Lax, Sam McVeety, Daniel Mills, Frances Perry, Eric Schmidt, 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), 2015.

[2] Muhammad Intizar Ali, Naomi Ono, Mahedi Kaysar, Zia Ush Shamszaman, Thu-Le Pham, Feng Gao, Keith Griffin, and Alessandra Mileo. Real-time data analytics and event detection for iot-enabled communication systems. *Journal of Web Semantics*, 42:19–37, 2017.

[3] Daniel Alvarez-Coello and Jorge Marx Gómez. Ontology-based integration of vehicle-related data. In *2021 IEEE 15th international conference on semantic computing (ICSC)*, pages 437–442. IEEE, 2021.

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

[5] Ahmed Awad, Riccardo Tommasini, Samuele Langhi, Mahmoud Kamel, Emanuele Della Valle, and Sherif Sakr. D^{2}ia: User-defined interval analytics on distributed streams. *Inf. Syst.*, 104:101679, 2022.

[6] Franz Baader, Ian Horrocks, Carsten Lutz, and Uli Sattler. *Introduction to description logic*. Cambridge University Press, 2017.

[7] Marco Balduini, Stefano Bocconi, Alessandro Bozzon, Emanuele Della Valle, Yi Huang, Jasper Oosterman, Themis Palpanas, Mikalai Tsytsarau, et al. A case study of active, continuous and predictive social media analytics for smart city. In *S4SC@ ISWC*, pages 31–46, 2014.

[8] Marco Balduini, Irene Celino, Daniele DellAglio, Emanuele Della Valle, Yi Huang, Tony Lee, Seon-Ho Kim, and Volker Tresp. Bottari: An augmented reality mobile application to deliver personalized and location-based recommendations by continuous analysis of social media streams. *Journal of Web Semantics*, 16:33–41, 2012.

[9] Marco Balduini, Emanuele Della Valle, Matteo Azzi, Roberto Larcher, Fabrizio Antonelli, and Paolo Ciuccarelli. Citysensing: Fusing city data for visual storytelling. *IEEE MultiMedia*, 22(3):44–53, 2015.

[10] Marco Balduini, Emanuele Della Valle, Daniele DellAglio, Mikalai Tsytsarau, Themis Palpanas, and Cristian Confalonieri. Social listening of city scale events using the streaming linked data framework. In *The Semantic Web–ISWC 2013: 12th International Semantic Web Conference, Sydney, NSW, Australia, October 21-25, 2013, Proceedings, Part II 12*, pages 1–16. Springer, 2013.

[11] Marco Balduini, Emanuele Della Valle, and Riccardo Tommasini. Sld revolution: A cheaper, faster yet more accurate streaming linked data framework. In *The Semantic Web: ESWC 2017 Satellite Events: ESWC 2017 Satellite Events, Portorož, Slovenia, May 28–June 1, 2017, Revised Selected Papers 14*, pages 263–279. Springer, 2017.

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

[13] Snehasis Banerjee, Debnath Mukherjee, and Prateep Misra. 'what affects me?' a smart public alert system based on stream reasoning. In *Proceedings of the 7th International Conference on Ubiquitous Information Management and Communication*, pages 1–10, 2013.

[14] Davide Francesco Barbieri, Daniele Braga, Stefano Ceri, Emanuele Della Valle, and Michael Grossniklaus. C-SPARQL: a continuous query language for RDF data streams. *Int. J. Semantic Comput.*, 4(1):3–25, 2010.

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

[16] Pieter Bonte and Femke Ongenae. Towards cascading reasoning for generic edge processing. 2023.

[17] Pieter Bonte, Femke Ongenae, and Filip De Turck. Subset reasoning for event-based systems. *IEEE Access*, 7:107533–107549, 2019.

[18] Pieter Bonte and Riccardo Tommasini. Streaming linked data: A survey on life cycle compliance. *Journal of Web Semantics*, page 100785, 2023.

[19] 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.

[20] Pieter Bonte, Riccardo Tommasini, Filip De Turck, Femke Ongenae, and Emanuele Della Valle. C-sprite: Efficient hierarchical reasoning for rapid RDF stream processing. In *Proceedings of the 13th ACM International Conference on Distributed and Event-based Systems, DEBS 2019, Darmstadt, Germany, June 24-28, 2019*, pages 103–114. ACM, 2019.

[21] Pieter Bonte, Filip De Turck, and Femke Ongenae. Bridging the gap between expressivity and efficiency in stream reasoning: a structural caching approach for iot streams. *Knowledge and Information Systems*, 64(7):1781–1815, 2022.

[22] Irina Botan, Roozbeh Derakhshan, Nihal Dindar, Laura Haas, Renée J Miller, and Nesime Tatbul. Secret: a model for analysis of the execution semantics of stream processing systems. *Proceedings of the VLDB Endowment*, 3(1-2):232–243, 2010.

[23] 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.

[24] 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.

[25] Jean-Paul Calbimonte, Sofiane Sarni, Julien Eberle, and Karl Aberer. Xgsn: An open-source semantic sensing middleware for the web of things. In *TC/SSN@ ISWC*, pages 51–66, 2014.

[26] Davide Calvaresi and Jean-Paul Calbimonte. Real-time compliant stream processing agents for physical rehabilitation. *Sensors*, 20(3):746, 2020.

[27] Mathias De Brouwer, Femke Ongenae, Pieter Bonte, and Filip De Turck. Towards a cascading reasoning framework to support responsive ambient-intelligent healthcare interventions. *Sensors*, 18(10):3514, 2018.

[28] Mathias De Brouwer, Bram Steenwinckel, Ziye Fang, Marija Stojchevska, Pieter Bonte, Filip De Turck, Sofie Van Hoecke, and Femke Ongenae. Context-aware query derivation for iot data streams with divide enabling privacy by design. *Semantic Web Journal*, 2023.

[29] Mathias De Brouwer, Nicolas Vandenbussche, Bram Steenwinckel, Marija Stojchevska, Jonas Van Der Donckt, Vic Degraeve, Filip De Turck, Koen Paemeleire, Sofie Van Hoecke, and Femke Ongenae. Towards knowledge-driven symptom monitoring & trigger detection of primary headache disorders. In *Companion Proceedings of the Web Conference 2022*, pages 264–268, 2022.

[30] 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.

[31] 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.

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

[33] Giuseppe DAniello, Matteo Gaeta, and Francesco Orciuoli. An approach based on semantic stream reasoning to support decision processes in smart cities. *Telematics and Informatics*, 35(1):68–81, 2018.

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

[35] Javier D Fernández, Alejandro Llaves, and Oscar Corcho. Efficient rdf interchange (eri) format for rdf data streams. In *The Semantic Web–ISWC 2014: 13th International Semantic Web*

Conference, Riva del Garda, Italy, October 19-23, 2014. Proceedings, Part II 13, pages 244–259. Springer, 2014.

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

[37] 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.

[38] Valentinas Janeiko, Roonak Rezvani, Narges Pourshahrokhi, Shirin Enshaeifar, Marianne Krogbæk, Sebastian Holmgard Christophersen, Tarek Elsaleh, and Payam Barnaghi. Enabling context-aware search using extracted insights from iot data streams. In *2020 Global Internet of Things Summit (GIoTS)*, pages 1–6. IEEE, 2020.

[39] Andreas Kamilaris, Feng Gao, Francesc X Prenafeta-Boldu, and Muhammad Intizar Ali. Agri-iot: A semantic framework for internet of things-enabled smart farming applications. In *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*, pages 442–447. IEEE, 2016.

[40] Evgeny Kharlamov, Yannis Kotidis, Theofilos Mailis, Christian Neuenstadt, Charalampos Nikolaou, Özgür Özçep, Christoforos Svingos, Dmitriy Zheleznyakov, Sebastian Brandt, Ian Horrocks, et al. Towards analytics aware ontology based access to static and streaming data. In *The Semantic Web–ISWC 2016: 15th International Semantic Web Conference, Kobe, Japan, October 17–21, 2016, Proceedings, Part II 15*, pages 344–362. Springer, 2016.

[41] Evgeny Kharlamov, Theofilos Mailis, Gulnar Mehdi, Christian Neuenstadt, Özgür Özçep, Mikhail Roshchin, Nina Solomakhina, Ahmet Soylu, Christoforos Svingos, Sebastian Brandt, et al. Semantic access to streaming and static data at siemens. *Journal of Web Semantics*, 44:54–74, 2017.

[42] Evgeny Kharlamov, Nina Solomakhina, Özgür Lütfü Özçep, Dmitriy Zheleznyakov, Thomas Hubauer, Steffen Lamparter, Mikhail Roshchin, Ahmet Soylu, and Stuart Watson. How semantic technologies can enhance data access at siemens energy. In *The Semantic Web–ISWC 2014: 13th International Semantic Web Conference, Riva del Garda, Italy, October 19-23, 2014. Proceedings, Part I 13*, pages 601–619. Springer, 2014.

[43] 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.

[44] Srdjan Komazec, Davide Cerri, and Dieter Fensel. Sparkwave: continuous schema-enhanced pattern matching over rdf data streams. In *Proceedings of the 6th ACM International Conference on Distributed Event-Based Systems*, pages 58–68, 2012.

[45] Danh Le-Phuoc, Hoan Quoc Nguyen-Mau, Josiane Xavier Parreira, and Manfred Hauswirth. A middleware framework for scalable management of linked streams. *Journal of Web Semantics*, 16:42–51, 2012.

[46] Danh Le-Phuoc, Hoan Nguyen Mau Quoc, Josiane Xavier Parreira, and Manfred Hauswirth. The linked sensor middleware–connecting the real world and the semantic web. *Proceedings of the Semantic Web Challenge*, 152:22–23, 2011.

[47] Danh Le-Phuoc, Hoan Nguyen Mau Quoc, Hung Ngo Quoc, Tuan Tran Nhat, and Manfred Hauswirth. The graph of things: A step towards the live knowledge graph of connected things. *Journal of Web Semantics*, 37:25–35, 2016.

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

[49] Georgios Meditskos and Ioannis Kompatsiaris. iknow: Ontology-driven situational awareness for the recognition of activities of daily living. *Pervasive and Mobile Computing*, 40:17–41, 2017.

[50] Manh Nguyen-Duc, Anh Le-Tuan, Jean-Paul Calbimonte, Manfred Hauswirth, and Danh Le-Phuoc. Autonomous rdf stream processing for iot edge devices. In *Semantic Technology: 9th Joint International Conference, JIST 2019, Hangzhou, China, November 25–27, 2019, Proceedings 9*, pages 304–319. Springer, 2020.

[51] Jack E Olson. *Data quality: the accuracy dimension*. Elsevier, 2003.

[52] Danh Le Phuoc, Minh Dao-Tran, Anh Lê Tuán, Manh Nguyen Duc, and Manfred Hauswirth. RDF stream processing with CQELS framework for real-time analysis. In Frank Eliassen and Roman Vitenberg, editors, *Proceedings of the 9th ACM International Conference on Distributed Event-Based Systems, DEBS '15, Oslo, Norway, June 29 - July 3, 2015*, pages 285–292. ACM, 2015.

[53] María Poveda-Villalón, Paola Espinoza-Arias, Daniel Garijo, and Óscar Corcho. Coming to terms with FAIR ontologies. In C. Maria Keet and Michel Dumontier, editors, *Knowledge Engineering and Knowledge Management - 22nd International Conference, EKAW 2020, Bolzano, Italy, September 16-20, 2020, Proceedings*, volume 12387 of *Lecture Notes in Computer Science*, pages 255–270. Springer, 2020.

[54] Dan Puiu, Payam Barnaghi, Ralf Tönjes, Daniel Kümper, Muhammad Intizar Ali, Alessandra Mileo, Josiane Xavier Parreira, Marten Fischer, Sefki Kolozali, Nazli Farajidavar, et al. Citypulse: Large scale data analytics framework for smart cities. *IEEE Access*, 4:1086–1108, 2016.

[55] Paula Reyero Lobo, Enrico Daga, Harith Alani, and Miriam Fernandez. Semantic web technologies and bias in artificial intelligence: A systematic literature review. *Semantic Web*, (Preprint):1–26, 2022.

[56] Luca Roffia, Paolo Azzoni, Cristiano Aguzzi, Fabio Viola, Francesco Antoniazzi, and Tullio Salmon Cinotti. Dynamic linked data: A sparql event processing architecture. *Future Internet*, 10(4):36, 2018.

[57] Patrik Schneider, Daniel Alvarez-Coello, Anh Le-Tuan, Manh Nguyen Duc, and Danh Le Phuoc. Stream reasoning playground. In Paul Groth, Maria-Esther Vidal, Fabian M. Suchanek, Pedro A. Szekely, Pavan Kapanipathi, Catia Pesquita, Hala Skaf-Molli, and Minna Tamper, editors, *The Semantic Web - 19th International Conference, ESWC 2022, Hersonissos, Crete, Greece, May 29 - June 2, 2022, Proceedings*, volume 13261 of *Lecture Notes in Computer Science*, pages 406–424. Springer, 2022.

[58] Martin Serrano, Hoan Nguyen Mau Quoc, Danh Le Phuoc, Manfred Hauswirth, John Soldatos, Nikos Kefalakis, Prem Prakash Jayaraman, and Arkady Zaslavsky. Defining the stack for service delivery models and interoperability in the internet of things: A practical case with openiot-vdk. *IEEE Journal on Selected Areas in Communications*, 33(4):676–689, 2015.

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

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

[61] Ruben Taelman, Riccardo Tommasini, Joachim Van Herwegen, Miel Vander Sande, Emanuele Della Valle, and Ruben Verborgh. On the semantics of tpf-qs towards publishing and querying rdf streams at web-scale. In Anna Fensel, Victor de Boer, Tassilo Pellegrini, Elmar Kiesling, Bernhard Haslhofer, Laura Hollink, and Alexander Schindler, editors, *Proceedings of the 14th International Conference on Semantic Systems, SEMANTiCS 2018, Vienna, Austria, September 10-13, 2018*, volume 137 of *Procedia Computer Science*, pages 43–54. Elsevier, 2018.

[62] 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.

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

[64] 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.

[65] Riccardo Tommasini, Pieter Bonte, Fabiano Spiga, and Emanuele Della Valle. *Streaming Linked Data: From Vision to Practice*. Springer Nature, 2023.

[66] 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.

[67] 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.

[68] Peter A. Tucker, David Maier, Tim Sheard, and Leonidas Fegaras. Exploiting punctuation semantics in continuous data streams. *IEEE Trans. Knowl. Data Eng.*, 15(3):555–568, 2003.

[69] Emanuele Della Valle, Stefano Ceri, Frank van Harmelen, and Dieter Fensel. It's a streaming world! reasoning upon rapidly changing information. *IEEE Intelligent Systems*, 24(6):83–89, 2009.

[70] Brecht Van de Vyvere, Pieter Colpaert, Erik Mannens, and Ruben Verborgh. Open traffic lights: a strategy for publishing and preserving traffic lights data. In *Companion Proceedings of the 2019 World Wide Web Conference*, pages 966–971, 2019.

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

[72] Arthur Vercruysse, Sitt Min Oo, and Pieter Colpaert. Describing a network of live datasets with the sds vocabulary. In *MEPDaW2022, Managing the Evolution and Preservation of the Data Web*, pages 1–6, 2022.

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

[74] 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.

[75] Ying Zhang, Minh-Duc Pham, Óscar Corcho, and Jean-Paul Calbimonte. Srbench: A streaming RDF/SPARQL benchmark. In Philippe Cudré-Mauroux, Jeff Heflin, Evren Sirin, Tania Tudorache, Jérôme Euzenat, Manfred Hauswirth, Josiane Xavier Parreira, Jim Hendler, Guus Schreiber, Abraham Bernstein, and Eva Blomqvist, editors, *The Semantic Web - ISWC 2012 - 11th International Semantic Web Conference, Boston, MA, USA, November 11-15, 2012, Proceedings, Part I*, volume 7649 of *Lecture Notes in Computer Science*, pages 641–657. Springer, 2012.