1-1-2020

Provenance-aware knowledge representation: A survey of data models and contextualized knowledge graphs

Leslie F. Sikos
Edith Cowan University

Dean Philp

10.1007/s41019-020-00118-0
Sikos, L. F., & Philp, D. (2020). Provenance-aware knowledge representation: A survey of data models and contextualized knowledge graphs. Data Science and Engineering. https://doi.org/10.1007/s41019-020-00118-0

Data Science and Engineering
https://doi.org/10.1007/s41019-020-00118-0

Provenance-Aware Knowledge Representation: A Survey of Data Models and Contextualized Knowledge Graphs

Leslie F. Sikos[1] · **Dean Philp**[2]

Received: 1 April 2019 / Revised: 13 February 2020 / Accepted: 3 March 2020

Abstract
Expressing machine-interpretable statements in the form of subject-predicate-object triples is a well-established practice for capturing semantics of structured data. However, the standard used for representing these triples, RDF, inherently lacks the mechanism to attach provenance data, which would be crucial to make automatically generated and/or processed data authoritative. This paper is a critical review of data models, annotation frameworks, knowledge organization systems, serialization syntaxes, and algebras that enable provenance-aware RDF statements. The various approaches are assessed in terms of standard compliance, formal semantics, tuple type, vocabulary term usage, blank nodes, provenance granularity, and scalability. This can be used to advance existing solutions and help implementers to select the most suitable approach (or a combination of approaches) for their applications. Moreover, the analysis of the mechanisms and their limitations highlighted in this paper can serve as the basis for novel approaches in RDF-powered applications with increasing provenance needs.

Keywords RDF provenance · Contextual knowledge graph · RDF reification alternatives · RDF data model

1 Introduction to RDF Provenance

The *Resource Description Framework (RDF)*[1] is a Semantic Web standard for formal knowledge representation, which can be used to efficiently manipulate and interchange machine-interpretable, structured data. Its data model is particularly powerful due to its syntax and semantics; RDF allows statements to be made in the form of subject-predicate-object triples, resulting in fixed-length dataset fields that are much easier to process than variable-length fields. Formally speaking, assume pairwise disjoint infinite sets of

1. Internationalized Resource Identifiers (IRIs, $\mathbb{I}$), i.e., sets of strings of Unicode characters of the form `scheme:[//[user:password@]host[:port]] [/]path[?query] [#fragment]` used to identify a resource,[2]
2. RDF literals ($\mathbb{L}$), which can be a) self-denoting plain literals $\mathbb{L}_P$ in the form `"<string>"(@<lang>)?`, where `<string>` is a string and `<lang>` is an optional language tag, or b) typed literals $\mathbb{L}_T$ of the form `"<string>"^^<datatype>`, where `<datatype>` is an IRI denoting a datatype according to a schema (e.g., XML Schema), and `<string>` is an element of the lexical space corresponding to the datatype, and
3. blank nodes ($\mathbb{B}$), i.e., unique anonymous resources that do not belong to either of the above sets.

A triple of the form $(s, p, o) \in (\mathbb{I} \cup \mathbb{B}) \times \mathbb{I} \times (\mathbb{I} \cup \mathbb{L} \cup \mathbb{B})$ is called an *RDF triple*, also known as an *RDF statement*, where s is the subject, p is the predicate, and o is the object.

The RDF data model is the canonicalization of a directed graph, offering compatibility with graph algorithms, such as graph traversal algorithms [1]. In addition, the RDF data model inherently supports basic inferences. Being modular, it allows fully parallelized data processing and can represent partial information. RDF is one of the primary graph-based data models that are well-utilized in fields that require data

[1] https://www.w3.org/RDF/

[2] https://tools.ietf.org/html/rfc3987

✉ Leslie F. Sikos
l.sikos@ecu.edu.au

Dean Philp
Dean.Philp@dst.defence.gov.au

[1] Edith Cowan University, 270 Joondalup Drive, Joondalup, WA 6027, Australia

[2] Defence Science and Technology Group, Third Ave, Edinburgh, SA 5111, Australia

Published online: 08 May 2020

fusion and/or aggregation from diverse data sources, such as cybersecurity [2].

RDF has a variety of serialization formats and syntaxes, including RDF/XML,[3] Turtle,[4] Notation3 (N3),[5] N-triples,[6] N-quads,[7] JSON-LD,[8] RDF/JSON,[9] RDFa,[10] and HTML5 Microdata.[11] These make it possible to express RDF statements differently for applications that require compatibility with XML, the most compact representation possible, define property values in website markup attributes, and so on [3].

RDF can be used to provide the uniform representation of knowledge for processing data from diverse web sources via syntactic and semantic interoperability. Moreover, RDF facilitates the partial or full automation of tasks that otherwise would have to be processed manually [4, 5].

These benefits make RDF appealing for a wide range of applications; however, RDF has shortcomings when it comes to encapsulating metadata to statements. With the proliferation of heterogeneous structured data sources, such as triplestores and LOD datasets, capturing *data provenance*,[12] i.e., the origin or source of data [7], and the technique used to extract it, is becoming more and more important, because it enables the verification of data, the assessment of reliability [8], the analysis of the processes that generated the data [9], decision support for areas such as cybersecurity [10, 11], cyberthreat intelligence [12], and cyber-situational awareness [13], and helps express trustworthiness [14, 15], uncertainty [16], and data quality [17]. Yet, the RDF data model does not have a built-in mechanism to attach provenance to triples or elements of triples.[13] Consequently, representing provenance data with RDF triples is a long-standing, non-trivial problem [19]. While the World Wide Web Consortium (W3C) suggested RDF extensions in 2010 to support provenance in the upcoming version of the standard [20, 21], none of these have been implemented in the next release in 2014, namely in RDF 1.1.[14] Related efforts of the W3C have been kept to a minimum with the shutdown of the Provenance Working Group in 2013 [22].

However, in Semantic Web applications, provenance can be seen as a means to develop trust [23], as witnessed by implementations of *augmented provenance* [24, 25] and *semantic provenance* [26, 27] in application areas such as eScience [28–32] and scientific data processing [33], workflow management, bioinformatics [34], laboratory information management [35], digital media archives [36], recommender systems [37], query search result ranking [38], and decision support for cybersecurity and cyber-situational awareness [39, 40].

This paper is a critical review of alternate approaches to capture provenance for RDF, demonstrates their use in, and provides their quantitative comparison for, the cybersecurity domain (which is known to be reliant on provenance) from various perspectives. Section 2 covers extensions of the standard RDF data model, state-of-the-art annotation frameworks, and purpose-built knowledge organization systems (KOS), including *controlled vocabularies*[15] and *ontologies*,[16] typically written in *RDFS*[17] and *OWL*,[18] respectively. Section 3 discusses the requirements of RDF triplestores, quadstores, and graph databases to be suitable for storing provenance-aware RDF data, and Sect. 4 details how RDF provenance can be queried. Section 5 reviews the most prominent software tools for manipulating RDF provenance. Section 6 describes common application domains. Finally, Sect. 7 demonstrates how application-specific implementations can outperform general-purpose RDF provenance techniques.

2 Formal Representation of RDF Data Provenance

RDF reification[19] refers to making an RDF statement about another RDF statement by instantiating the `rdf:`

[3] https://www.w3.org/TR/rdf-syntax-grammar/

[4] https://www.w3.org/TR/turtle/

[5] https://www.w3.org/TeamSubmission/n3/

[6] https://www.w3.org/TR/n-triples/

[7] https://www.w3.org/TR/n-quads/

[8] https://www.w3.org/TR/json-ld/

[9] https://www.w3.org/TR/rdf-json/

[10] https://www.w3.org/TR/rdfa-primer/

[11] https://www.w3.org/TR/microdata/

[12] Note that *provenance* is a complex term, and its usage varies greatly in different contexts [6].

[13] This can be circumvented by using alternate knowledge representations, such as considering entities as embeddings in a vector space, for example [18], however, none of the alternate representations share all the strengths of RDF.

[14] https://www.w3.org/TR/rdf11-new/

[15] A controlled vocabulary is a finite set of IRI symbols denoting concept names or classes (atomic concepts), role names, properties, and relationships (atomic roles), and individual names (entities), where these three sets are pairwise disjoint.

[16] Ontologies are formal conceptualizations of a knowledge domain with complex relationships, and optionally complex rules, suitable for inferring new statements, thereby making implicit knowledge explicit.

[17] *RDF Schema*, an extension of RDF's vocabulary for creating vocabularies, taxonomies, and thesauri; see https://www.w3.org/TR/rdf-schema/ for reference.

[18] *Web Ontology Language* (intentionally abbreviated with the W and O swapped as OWL), a fully featured knowledge representation language for the conceptualization of knowledge domains with complex property restrictions and concept relationships; see https://www.w3.org/OWL/ for reference.

[19] https://www.w3.org/TR/2004/REC-rdf-primer-20040210/#reification

`Statement` class and using the `rdf:subject`, `rdf:predicate`, and `rdf:object` properties of the standard RDF vocabulary[20] to identify the elements of the triple. It is the only standard syntax to capture RDF provenance and the only syntax with which all RDF systems are compatible. As an example, assume the RDF statement `"DREAMSCAPE-AS-AP" : hasASNumber "38719"`, expressing that the autonomous system number of Dreamscape Networks is 38719. By reifying this statement, its source can be captured as shown in Listing 1.

Listing 1 RDF reification

```
@prefix : <http://example.com/ > .
@prefix dc: <http://purl.org/dc/
    elements/1.1/> .
@prefix rdf: <http://www.w3.org
    /1999/02/22-rdf-syntax-ns#> .
_:x rdf:type rdf:Statement .
_:x rdf:subject "DREAMSCAPE-AS-AP"
    .
_:x rdf:predicate :hasASNumber .
_:x rdf:object "38719"^^xsd:integer
     .
_:x dc:source :APNIC .
```

The last statement captures the provenance of the original statement, namely that the source of the AS number is the Asia-Pacific Network Information Centre (APNIC).[21] However, the blank node (bnode) `_:x` used as part of the mechanism has no associated meaning and cannot be dereferenced globally.

Reified statements can describe not only the source of RDF triples, but also changes made to the structure of RDF graphs, for example, by referring to statements that have been amended in, added to, or removed from, an RDF dataset.

However, reification has no formal semantics, and leads to a high increase in the number of triples, hence, it does not scale well. After all, reification requires a statement about the subject, another statement about the predicate, and a third statement about the object of the triple, plus at least one more statement that captures provenance, i.e., the number of statements in the dataset will increase at least four times. This "triple bloat" is one of the main reasons for the unpopularity of reification.

Note that if the provenance triple is stored in the same RDF/XML file as the original statement, a shorthand notation can also be used. Instead of specifying the subject, predicate, and object of the original triple, the reified statement can be identified with the value of the `rdf:ID` property (see Listing 2).[22]

Listing 2 Shorthand notation for RDF reification

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.
    org/1999/02/22-rdf-syntax-ns#"
    xmlns:ex="http://example.com/">
<rdf:Description rdf:about="http://
    example.org/DREAMSCAPE-AS-AP/">
<ex:hasASNumber rdf:ID="DCAS">
    38719</ex:hasASNumber>
</rdf:Description>
<rdf:Description rdf:about=
 "https://www.apnic.net">
<ex:sourceOf rdf:resource="#DCAS"
    />
</rdf:Description>
</rdf:RDF>
```

The triple identified this way is recognized by RDF parsers, which then automatically annotate the subject, the predicate, and the object.

Nevertheless, reification has another shortcoming: writing queries to retrieve statement-level provenance data is cumbersome, because an additional subexpression has to complement the provenance-related subexpressions in queries to be able to match the reification triples. For these reasons, some proposed reification to be deprecated.[23,24]

The other approach suggested by the W3C to define additional attributes, including provenance, to RDF triples, is called *n-ary relations*, which provides a mechanism to describe the relationship of an individual with more than one other individual or data type value [41]. This is in contrast with the binary relations most common in Semantic Web languages, which link an entity either to another entity or to a datatype value, such as a string or an integer number.

To express our previous example with an n-ary relation, `hasASNumber` is defined as a property of the individual `DREAMSCAPE-AS-AP`, with another object (`_:AS_Relation_1`, an instance of the class `AS_Relation`) as its value (see Listing 3).

Listing 3 N-ary relation in RDF

```
:DREAMSCAPE-AS-AP a :AS ;
:hasASNumber _:AS_Relation_1 .
:AS_Relation_1 a :AS_Relation ;
:hasASNumber "38719";
:accordingTo :APNIC .
```

The individual `_:AS_Relation_1` represents a single object that encapsulates both the AS number (`38719`) and its source (`APNIC`), as shown in Fig. 1.

Notice the use of a (self-serving) blank node, which cannot be dereferenced globally, to represent instances of an n-ary RDF relation.

[20] https://www.w3.org/1999/02/22-rdf-syntax-ns.ttl

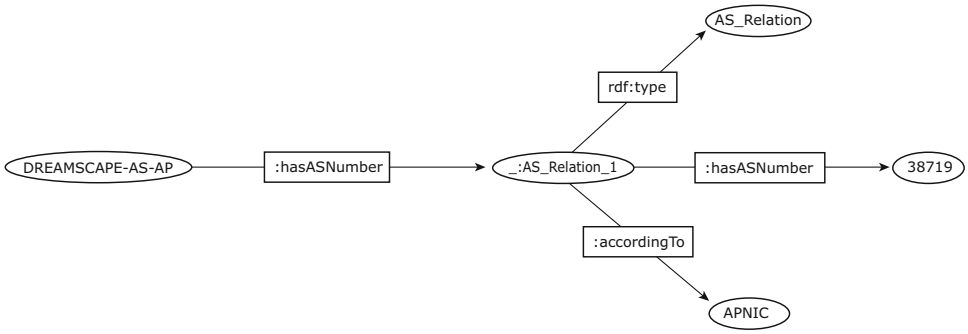
[21] https://www.apnic.net

[22] https://www.w3.org/TR/rdf-syntax-grammar/#section-Syntax-reifying.

[23] https://www.w3.org/2009/12/rdf-ws/papers/ws11

[24] https://lists.w3.org/Archives/Public/public-rdf-wg/2011Apr/0164.html

Fig. 1 An n-ary relationship



The implementation issues arising from the design of reification and n-ary relations, and the scalability limitations in particular, resulted in alternate approaches to provide provenance to RDF triples. The next sections provide a comprehensive systematic review of these approaches.

2.1 Data Models and Annotation Frameworks for RDF Provenance

Approaches to provide alternatives to RDF reification and n-ary relations include

- lossless decomposition of RDF graphs: *RDF Molecule* [42];
- extensions of the RDF data model: *N3Logic* [43,44], *RDF*$^+$ [45,46], *annotated RDF (aRDF)* [47], *SPO+Time+Location (SPOTL)* [48], *RDF** [49], RSP-QL* [50];
- alternate data models: mapping objects to vectors [18], *GSMM* [51];
- extensions of the RDFS semantics: *Annotated RDF Schema* [52,53];
- purpose-designed implementation techniques
  - using annotations: *RDF/XML Source Declaration*,[25] resource annotation [54];
  - via encapsulating provenance information in tuple elements: *Provenance Context Entity (PaCE)* [55], *Singleton property* [56];
  - using knowledge organization systems;
- adding provenance to triples, forming RDF quadruples: *N-Quads*,[26] *Named graphs* [57,58], *RDF/S graphsets* [59], *RDF triple coloring* [60], *nanopublications* [61], *Hoganification* [62]), *GraphSource* (Sikos et al. [40] and Sikos et al. [63] collectively);
- hybrid approaches, which have multiple traits of the above categories, such as *g-RDF* [64], which extends RDFS semantics, defines provenance stable models and provenance Herbrand interpretations, and utilizes ontologies with positive and strongly negated RDF triples (gRDF triples) and derivation rules.

Approaches have also been developed for capturing other types of metadata, such as temporal constraints, for RDF, some of which may be suitable for capturing provenance as well (e.g., *temporal RDF* [65–67]).

The reasoning potential of provenance representations can be improved by using OWL and SWRL reasoning rules [68], extending OWL to be able to define contexts-dependent axioms [69], and purpose-designed context-aware reasoning rules that are not restricted by the limitations of Semantic Web languages [39].[27]

The abstraction of reification techniques with description logics led to the introduction of *contextual annotation*, which focuses on the logical formalism behind provenance-aware statements rather than on the data model [70]. Moreover, a family of DLs has been proposed specifically for representing data provenance [71].

To demonstrate the fundamental differences, let us briefly describe our running example with various approaches.

The RDF/XML syntax extension can turn triples into quadruples, where the forth tuple element is the IRI of the source of the triple, defined as the attribute of `cos:graph` (see Listing 4).

Listing 4 RDF/XML Source Declaration

```
<rdf:RDF
xmlns:ex="http://example.com/"
xmlns:rdf="http://www.w3.org/
    1999/02/22-rdf-syntax-ns#"
xmlns:cos="http://www.inria.fr/
    acacia/corese#"
cos:graph="http://www.apnic.net">
```

[25] https://www.w3.org/Submission/rdfsource/

[26] https://www.w3.org/TR/n-quads/

[27] Because the context identifiers of RDF quadruples have no restrictions regarding what they represent, there is no general entailment regime for the context element in quad-based provenance-aware statements, and therefore, provenance-related inferences rely on the triple elements in both triple-based and quad-based representations. Although writing custom, application-specific rulesets could provide far more sophisticated reasoning capabilities than what is possible with standard RDFS and OWL entailment and reasoning over standard provenance ontologies, such rulesets are yet to be developed.

Table 1 An RDF$^+$ quintuple

Subject	Predicate	Object	Meta-property	Meta-value
DREAMSCAPE-AS-AP	hasASNumber	38719	accordingTo	APNIC

```
<rdf:Description rdf:about="http://
    example.com/DREAMSCAPE/">
<ex:hasASNumber>38719
  </ex:hasASNumber>
</rdf:Description>
</rdf:RDF>
```

An aRDF triple consists of an ordinary RDF triple and its annotation (see Listing 5).

Listing 5 Annotated RDF (aRDF)

```
(APNIC, states: (hasASNumber,
    38719), DREAMSCAPE-AS-AP)
```

By using RDF molecules, RDF graphs can be decomposed to their finest, lossless subgraphs. The decomposition can be performed by identifying blank nodes in RDF graphs that connect triples (naïve decomposition), use functional dependency semantics (functional decomposition), or use extended functional dependency (heuristic decomposition). For example, a triple from a network knowledge discovery dataset and another triple from a dataset of APNIC autonomous systems can form an RDF molecule (see Listing 6).

Listing 6 An RDF molecule

```
{t1} (:ASDS :hasASname "DREAMSCAPE-
    AS-AP" )
{t2} ("DREAMSCAPE-AS-AP"
  :hasASNumber "38719" )
```

In N3Logic, provenance can be captured in the form of *quoted N3 formulae*, for example the one shown in Listing 7.

Listing 7 A quoted N3 formula

```
:APNIC :states { "DREAMSCAPE-AS-AP"
    :hasASNumber "38719" } .
```

RDF$^+$ extends RDF with provenance data support by attaching a metadata property and its value to each triple (see Table 1).

SPOTL base facts can be written with semantic meta-facts as demonstrated in Listing 8.

Listing 8 A SPOTL base fact with a semantic meta-fact

```
#42: "DREAMSCAPE-AS-AP"
  :hasASNumber "38719"
#43: #42:  accordingTo: APNIC
```

RDF* encloses *embedded triples* between << and >> in the Turtle extension Turtle* (see Listing 9).

Listing 9 An embedded triple in RDF*

```
<<"DREAMSCAPE-AS-AP" :hasASNumber
    "38719">> :accordingTo
:APNIC .
```

Listing 10 shows the same statement using Annotated RDF Schema.

Listing 10 Annotated RDF Schema

```
("DREAMSCAPE-AS-AP" :hasASNumber
    "38719") : [:accordingTo,
:APNIC]
```

Our running example can be expressed using PaCE as shown in Listing 11 and with a singleton property as shown in Listing 12.

Listing 11 PaCE example

```
:DREAMSCAPE-AS-AP_APNIC a
 :DREAMSCAPE-AS-AP .
:DREAMSCAPE-AS-AP_APNIC
 :hasASNumber "38719" .
:DREAMSCAPE-AS-AP_APNIC
  :accordingTo :APNIC .
```

Listing 12 A singleton property example

```
"DREAMSCAPE-AS-AP" :hasASNumber#1
    "38719" .
:hasASNumber#1
  rdf:singletonPropertyOf
  :hasASNumber ;
:accordingTo :APNIC .
```

N-Quads inherently has a fourth column, which can be used, among other things, to declare provenance (see Listing 13).

Listing 13 N-Quads example

```
"DREAMSCAPE-AS-AP" :hasASNumber
    "38719" _:prov .
_:prov :source :accordingTo
: APNIC .
```

By using two named graphs, an RDF graph can describe the RDF statements (default graph or assertion graph), and another graph can detail the provenance data (provenance graph), as demonstrated in Listing 14.

Listing 14 Named graphs (GraphSource implementation)

```
ASDATA1 { "DREAMSCAPE-AS-AP"
 :hasASNumber "38719" . }

PROVENANCE { <http://example.org/
    ntwknowledge/>
 prov:wasDerivedFrom
 :BGP143R1to221R1 . }
```

RDF triple coloring can capture data integration scenarios in which the same data were derived from different resources. For example, assume the scenario shown in Table 2.

Based on the statements in Table 2, the triple in Listing 15 can be inferred.

Table 2 RDF triple coloring

s	*p*	*o*	"Color"
:APNICassigned	rdf:type	owl:Class	c_1
:APNICassigned	rdfs:subclassOf	:RIRassigned	c_1
:RIRassigned	rdfs:subclassOf	:IANAregistered	c_2

Listing 15 Statement inferred from the statements of Table 2

```
:APNICassigned
 rdfs:subclassOf :IANAregistered .
```

In this case, to define the origin of the statement, both c_1 and c_2 have to be assigned (see Listing 16).

Listing 16 Expressing composite origin with two triple colors

```
:APNICassigned rdfs:subclassOf :RIRassigned . ⟶ c1
:RIRassigned rdfs:subclassOf :IANAregistered . ⟶ c2
```

Otherwise, the composite nature of the data provenance would not be captured, because querying the triples with color c_1 would return `:APNICassigned rdfs:subclassOf :IANAregistered`, even though this is based on both c_1 and c_2, not just c_1. RDF triple coloring can capture implicit triples using colors defined by the + operation, such as $c_{1,2} = c_1 + c_2$. In this case, $c_{1,2}$ is a new URI assigned to those triples that are implied by triples colored c_1 and c_2.

Nanopublications can be efficiently implemented in the form of named graphs, as demonstrated in Listing 17.

Listing 17 A nanopublication

```
@prefix : <http://www.example.org/
    networkdataset180322#> .
@prefix prov: <http://www.w3.org/ns
    /prov#> .
:NETW { "DREAMSCAPE-AS-AP"
 :hasASNumber "38719" . }
:PROV { :NETW prov:wasDerivedFrom
 :APNIC . }
:META { :PROV prov:generatedAtTime
"2018-03-22T15:49:00+09:30"^^xsd:
    dateTime . }
```

For the detailed comparison of these approaches, the following aspects have been considered: availability of formal semantics, tuple type (triple, quad, or quintuple), compliance with the standard RDF data model and standard SPARQL algebra, reliance on external vocabularies, utilization of blank nodes, the granularity of data provenance that can be captured, and scalability.

2.1.1 Formal Semantics

The definition of Tarski-style model-theoretic semantics [72] for RDF graphs and the RDF and RDFS vocabularies provides a formal specification of when truth is preserved by RDF transformations and operations that derive RDF content from other RDF resources [73]. This is why defining formal semantics is a fundamental requirement for RDF reification approaches.

The semantics of RDF graphs is fixed via interpretations, such as *simple interpretations* and *RDF interpretations* [74]. A simple interpretation I is defined as follows [75]. IR is a nonempty set of resources, called the domain or universe of discourse of I; IP is the set of generic properties of I; I_{EXT} is a function that assigns to each property a set of pairs from IR, i.e., I_{EXT}: IP $\rightarrow 2^{IR \times IR}$, where $I_{EXT}(p)$ is called the extension of property p; I_s is a function that maps IRIs from V into the union set of IR and IP; IL is a function mapping the typed literals from V into the set of resources R; and LV is a subset of IR (the set of literal values). RDF interpretations have to satisfy additional semantic conditions on `xsd:string` and part of the infinite set of IRIs with the namespace prefix.

An *RDFS interpretation* is an RDF interpretation, which satisfies additional semantic conditions outlined by Hayes and Patel-Schneider [76]. $IC_{EXT}(y)$ is defined to be {x : ⟨x,y⟩ is in I_{EXT}(I(`rdf:type`))}. IC is defined to be IC_{EXT}(I(`rdfs:Class`)). By definition, LV is IC_{EXT} (I(`rdfs:Literal`)). IC_{EXT}(I(`rdfs:Resource`)) = IR. IC_{EXT}(I(`rdf:langString`)) is the set I(E), where E is a language-tagged string. For every other IRI *aaa* in D, IC_{EXT}(I(*aaa*)) is the value space of I(*aaa*) and for every IRI *aaa* in D, I(*aaa*) is in IC_{EXT}(I(`rdfs:Datatype`)). If ⟨*x*,*y*⟩ is in I_{EXT}(I(`rdfs:domain`)) and ⟨*u*,*v*⟩ is in $I_{EXT}(x)$ then *u* is in $IC_{EXT}(y)$. If ⟨*x*,*y*⟩ is in I_{EXT}(I(`rdfs:range`)) and ⟨*u*,*v*⟩ is in $I_{EXT}(x)$, then *v* is in $IC_{EXT}(y)$. I_{EXT}(I(`rdfs:subPropertyOf`)) is transitive and reflexive on IP. If ⟨*x*,*y*⟩ is in I_{EXT}(I(`rdfs:subPropertyOf`)), then *x* and *y* are in IP and $I_{EXT}(x)$ is a subset of $I_{EXT}(y)$. If *x* is in IC, then ⟨*x*, I(rdfs:Resource)⟩ is in I_{EXT}(I(`rdfs:subClassOf`)). I_{EXT}(I(`rdfs:subClassOf`)) is transitive and reflexive on IC. If ⟨*x*,*y*⟩ is in I_{EXT}(I(`rdfs:subClassOf`)), then *x* and *y* are in IC and $IC_{EXT}(x)$ is a subset of $IC_{EXT}(y)$. If *x* is in IC_{EXT}(I(`rdfs:ContainerMembershipProperty`)), then ⟨*x*, I(rdfs:member)⟩ is in I_{EXT}(I(`rdfs:subPropertyOf`)). If *x* is in IC_{EXT}(I(`rdfs:Datatype`)), then ⟨*x*, I(`rdfs:Literal`)⟩ is in I_{EXT}(I(`rdfs:subClassOf`)).

In addition, an RDFS interpretation also has to satisfy all the RDFS axiomatic triples.[28]

Among the RDF reification alternatives that define model-theoretic semantics, PaCE, the singleton property approach,

[28] https://www.w3.org/TR/rdf11-mt/#RDFS_axiomatic_triples

Table 3 Formal semantics of reification alternatives

Approach	Formal semantics
RDF/XML Source Declaration	Inherently (RDF semantics)
aRDF	Formal declarative semantics (model theory)
RDF Molecule	–
N3Logic	Model-theoretic semantics
RDF$^+$	Model-theoretic semantics
SPOTL	An extension of the standard RDF triple model
RDF*	Model-theoretic semantics (allows RDF* graphs to be transformed to standard RDF graphs)
Annotated RDFS	An extension of RDFS semantics
PaCE	Model-theoretic semantics extending standard RDFS semantics with additional condition (meta-rule)
Singleton property	Model-theoretic semantics; extension of simple and RDF interpretation
N-Quads	Inherently (RDF semantics)
Named graphs (incl. GraphSource)	Model-theoretic semantics; simple semantic extension of RDF(S) semantics
RDF/S graphsets	Model-theoretic semantics (generalizes named graphs)
RDF triple coloring	Coherence semantics [77]
Nanopublications	Depends on the implementation (model-theoretic semantics when implemented as named graphs)

and named graphs define their semantics as extensions of these standard semantics (see Table 3).

The aRDF declarative semantics are defined as follows. An aRDF interpretation I is a mapping from *Univ* to $\mathcal{A}$, where $\mathcal{A}$ is a partial order, and satisfies $(r, p{:}a, v)$ iff $a \preccurlyeq I(r,p,v)$. I satisfies an aRDF theory iff I satisfies every $(r,p{:}a,v) \in O$ and for all transitive properties $p \in \mathcal{P}$ and for all p-paths $Q = \{t_1, \ldots, t_k\}$ in O, where $t_i = (r_i, p_i : a_i, r_{i+1})$, and for all $a \in \mathcal{A}$ such that $a \preccurlyeq a_i$ for all $1 \leqslant i \leqslant k$, it is the case that $a \preccurlyeq I(r_1, p, r_{k+1})$. O is consistent iff there is at least one aRDF interpretation that satisfies it. O entails $(r,p{:}a,v)$ iff every aRDF interpretation that satisfies O also satisfies $(r,p{:}a,v)$.

The model-theoretic semantics of PaCE is an extension of RDFS semantics and can be defined as follows. Let provenance context pc of an RDF triple $\alpha = (S,P,O)$ be a common object of the predicate `provenir:derives_from`[29] associated with the triple. An RDFS-PaCE interpretation $\mathcal{I}$ of a vocabulary V is defined as an RDFS interpretation of the vocabulary $V \cup V_{PaCE}$ satisfying the additional condition (meta-rule) that for RDF triples $\alpha = (S_1,P_1,O_1)$ and $\beta = (S_2,P_2,O_2)$, provenance-determined predicates (that are specified to the application domain), and entities v, if $pc(\alpha) = pc(\beta)$, then $(S_1,p,v) = (S_2,p,v)$, $(P_1,p,v) = (P_2,p,v)$, and $(O_1,p,v) = (O_2,p,v)$. A graph G_1 PaCE-entails a graph G_2 if every RDFS-PaCE interpretation that is a model of G_1 is also a model of G_2. All inferences that can be made using simple, RDF, or RDFS entailment are also PaCE entailments.

The model-theoretic semantics of the singleton property approach extends a simple interpretation I to satisfy the following additional criteria: IP_S is a subset of IR, called the set of singleton properties of I, and $I_{S\_EXT}(p_s)$ is a function assigning to each singleton property a pair of entities from IR, formally $I_{S\_EXT}{:}IP_s \rightarrow IR \times IR$. As for an RDF interpretation $\mathcal{I}$, the semantics of the singleton property approach defines the following additional criteria: $x_s \in IP_s$ if $\langle x_s, \texttt{rdf:singletonPropertyOf}^{\mathcal{I}}\rangle \in I_{EXT}(\texttt{rdf:type}^{\mathcal{I}})$, $x_s \in IP_s$ if $\langle x_s,x^{\mathcal{I}}\rangle \in I_{EXT}(\texttt{rdf:singletonPropertyOf}^{\mathcal{I}})$, and $x \in IP$, $I_{S\_EXT}(x_s) = \langle s_1,s_2\rangle$.

The semantics of named graphs extends the RDF(S) semantics. An RDF(S) interpretation I conforms to a set of named graphs N if for every named graph $ng \in N$, name(ng) is in the vocabulary of I and I(name(ng)) = ng. While named graphs can attach metadata to a set of triples, they may have ambiguous semantics while associating different types of metadata at the triple level.

The semantics of Annotated RDFS can be defined as follows. An annotation domain for RDFS is an idempotent, commutative semiring of the form $D = \langle L, \oplus, \otimes, \bot, \top\rangle$, where L a nonempty set of annotation values and $\oplus$ is $\top$-annihilating.[30] Being an idempotent semi-ring, an annotation domain D induces a partial order $\preccurlyeq$ over L defined as $\lambda_1 \preccurlyeq \lambda_2$ iff $\lambda_1 \oplus \lambda_2 = \lambda_2$, which is suitable for expressing entailed or subsumed information. An annotated interpretation $\mathcal{I}$ over a vocabulary V is a tuple $\mathcal{I} = \langle \Delta_R, \Delta_P, \Delta_C, \Delta_L, P^{\mathcal{I}}[\![.]\!], C^{\mathcal{I}}[\![.]\!]\rangle$, where Δ_R, Δ_P, Δ_C, Δ_L

[29] The `provenir` prefix abbreviates the now-discontinued ontology URL http://knoesis1.wright.edu/library/ontologies/provenir/provenir.owl.

[30] For $\lambda, \lambda_i \in L$, $\oplus$ is idempotent, commutative, and associative; $\otimes$ is commutative and associative; $\bot \oplus \lambda = \lambda$, $\top \otimes \lambda = \lambda$, $\bot \otimes \lambda = \lambda$, and $\top \oplus \lambda = \top$; $\otimes$ is a distributive order over $\oplus$, i.e., $\lambda_1 \otimes (\lambda_2 \oplus \lambda_3) = (\lambda_1 \otimes \lambda_2) \oplus (\lambda_1 \otimes \lambda_3)$.

Table 4 Reification alternatives employ 3–6 elements per statement

Approach	Tuple type
RDF/XML Source Declaration	Triple (in RDF/XML)—implies quadruple
aRDF	Nonstandard
RDF Molecule	Quadruple
N3Logic	Triple (in N3)
RDF$^+$	Quintuple
SPOTL	Quadruple/nonstandard (quintuple/sextuple)
RDF*	Nonstandard (“metadata triple”)
Annotated RDFS	Nonstandard (“annotated triple”)
PaCE	Triple
Singleton property	Triple
N-Quads	Quadruple
Named graphs (incl. GraphSource)	Quadruple
RDF/S graphsets	Quadruple
RDF triple coloring	Quadruple
Nanopublications	Quadruple

are interpretation domains of $\mathcal{I}$ and $P[\![.]\!]$, $C[\![.]\!]$, $\cdot^{\mathcal{I}}$ are interpretation functions of $\mathcal{I}$. Δ_R is a nonempty finite set of resources (the domain of $\mathcal{I}$), Δ_P is a finite set of property names (not necessarily disjoint from Δ_R), $\Delta_C \subseteq \Delta_R$ is a distinguished subset of Δ_R) identifying if a resource denotes a class of resources, $\Delta_L \subseteq \Delta_R$ the set if literal values, Δ_L contains all plain literals in $L \cap V$, $P[\![.]\!]$ maps each property name $p \in \Delta_P$ into a function $P[\![p]\!] : \Delta_R \times \Delta_R \rightarrow L$, i.e., assigns an annotation value to each pair of resources; $C[\![.]\!]$ maps each class $c \in \Delta_C$ into a function $C[\![c]\!] : \Delta_R \rightarrow L$, i.e., assigns an annotation value representing class membership in c to every resource; $\cdot^{\mathcal{I}}$ maps each $t \in U_L \cap V$ into a value $t^{\mathcal{I}} \in \Delta_R \cup \Delta_P$ and such that $\cdot^{\mathcal{I}}$ is the identity for plain literals and assigns an element in Δ_R to each element in L.

2.1.2 Tuple Type

RDF reification alternatives use either standard RDF triples or quads, or nonstandard tuples to capture provenance (see Table 4).

RDF/XML Source Declaration employs standard RDF triples written in RDF/XML serialization. PaCE uses standard triples with carefully named entities to describe provenance-aware facts by indicating the data source in subjects and objects. In contrast, the singleton property approach uses standard RDF triples to capture provenance with the predicate; the instantiation triples define singleton properties as singleton properties of the base predicates, which can be used as predicates in singleton triples and as subjects in metadata triples.

N-Quads, named graphs, RDF triple coloring, RDF/S graphsets, and nanopublications use standard RDF quadruples. Annotated RDFS defines a proprietary tuple that, depending on the application, may be expressed using standard RDF quads. RDF$^+$ is the only approach that employs quintuples to capture provenance. aRDF and RDF* define tuples that are not compatible with RDF triples or quads.

The number of tuple elements for SPOTL depends on the implementation. When implemented in named graphs, SPOTL uses quadruples. RDF triples extended with temporal and spatial information can be written in quintuples, which can be further extended into sextuples with context or provenance information (the latter is called SPOTLX).

2.1.3 Standard Compliance

RDF reification alternatives include various data models, some of which extend the standard RDF data model [17]. The RDF/XML Source Declaration is compliant with RDF and SPARQL, although it has some minor implementation prerequisites, and is bound to a single serialization format. RDF molecules are decompositions of standard RDF graphs, and may be implemented in standard quadruple serializations. Only its advanced version is compatible with standard SPARQL queries [78]. N3Logic is a minimal extension of the standard RDF data model. It is natively supported by the standard N3 serialization. RDF$^+$ extends the standard RDF data model, and requires mapping from/to RDF. It has a limited downward compatibility with the standard RDF data model in applications that ignore RDF$^+$ extensions not supported by RDF. Due to its significant divergence from standard RDF, it cannot be expressed in any standard RDF serialization. Furthermore, it extends the standard SPARQL syntax and semantics, making it necessary to map queries. SPOTL extends the triple-based RDF data model to support additional data for statements. Depending on the implementation, it may or may not be implemented in named graphs. RDF* extends the standard RDF data model and requires

Table 5 Not all reification alternatives are standard-compliant

Approach	Compliance with standard		
	RDF data model	RDF serializations	SPARQL algebra
RDF/XML Source Declaration	+	RDF/XML only	+
aRDF	–	–	–
RDF Molecule	+	TriG, TriX, N-Quads	–/+
N3Logic	–	N3	+
RDF$^+$	–	–	–
SPOTL	–	–/+ (quadruple serializations)	–
RDF*	–	–	–
Annotated RDFS	–	– (potentially TriG/TriX/N-Quads)	–
PaCE	+	RDF/XML, N3, Turtle, N-Triples, RDF-JSON, JSON-LD, RDFa, HTML5 Microdata	+
Singleton property	+	RDF/XML, N3, Turtle, N-Triples, RDF-JSON, JSON-LD, RDFa, HTML5 Microdata	+
N-Quads	+	N-Quads	+
Named graphs (incl. GraphSource)	+	TriG, TriX, N-Quads	+
RDF/S graphsets	–	TriG, TriX, N-Quads	–
RDF triple coloring	+	TriG, TriX, N-Quads	+
Nanopublications	+	TriG, TriX	+

mapping. It proposes a proprietary extension to Turtle, called *Turtle**, in which triples are embedded in other triples. However, it cannot be directly implemented in any standard serialization. For querying, it extends the standard SPARQL algebra to *SPARQL**, which requires mapping. Annotated RDFS extends the standard RDF data model, although it may be expressed using standard quadruple serializations. Moreover, it extends the standard SPARQL algebra to "*Annotated SPARQL*" (*AnQL*). The PaCE approach is compatible with the standard RDF data model, and can be expressed using standard triple serialization formats. Similarly, the singleton property approach is RDF-compatible and can be written in any standard triple serialization. The singleton property approach is compatible with the standard SPARQL algebra, and allows the utilization of three types of triples in graph patterns: the statement about the instantiating singleton property, the singleton triple, and the metadata triple. N-Quads and named graphs are standard-compliant approaches. The RDF/S graphsets approach extends the RDFS data model and is not compatible with SPARQL: It extends the RDF query language (RQL) instead. Table 5 summarizes these approaches in terms of standard compliance.

2.1.4 Reliance on External Vocabularies

Some approaches rely on external vocabularies as part of their mechanism to capture provenance for RDF data (see Table 6).

RDF/XML Source Declaration utilizes the `cos:graph` attribute[31] in standard RDF/XML documents. RDF molecules do not rely on external vocabularies, however, for fine decompositions, a background ontology is required. N3Logic extends RDF with a vocabulary of predicates by reusing terms from the `log`,[32] `crypto`,[33] `list`,[34] `math`,[35] `os`,[36] `string`,[37] and `time`[38] namespaces. The nanopublications approach defines an ontology, called the Nanopublication Ontology,[39] but its use is not essential for implementing nanopublications. The application-specific GraphSource

[31] The `cos:` prefix abbreviates the now-discontinued ontology IRI http://www.inria.fr/acacia/corese#.

[32] http://www.w3.org/2000/10/swap/log#

[33] http://www.w3.org/2000/10/swap/crypto#

[34] http://www.w3.org/2000/10/swap/list#

[35] http://www.w3.org/2000/10/swap/math#

[36] http://www.w3.org/2000/10/swap/os#

[37] http://www.w3.org/2000/10/swap/string#

[38] http://www.w3.org/2000/10/swap/time#

[39] http://www.nanopub.org/nschema

Table 6 Some approaches rely on a single term or an entire ontology to capture provenance

Approach	Proprietary term or external vocabulary
RDF/XML Source Declaration	`cos:graph` attribute
aRDF	–
RDF Molecule	–
N3Logic	N3Logic Vocabulary
RDF$^+$	–
SPOTL	–
RDF*	–
Annotated RDFS	–
PaCE	PROVENIR Ontology (proprietary)
Singleton property	Nonstandard extension of rdfV with the `singletonPropertyOf` property
N-Quads	–
Named graphs (incl. GraphSource)	–
RDF/S graphsets	–
RDF triple coloring	–
Nanopublications	–

approach does not have a prerequisite for ontologies, but it is most efficient when used with the *Communication Network Topology and Forwarding Ontology (CNTFO)*.[40]

In contrast to these, there are approaches that constitute ontological models, such as *resource annotation* [54], which associates a single `rdfs:Resource` with a target, and utilizes domain ontology terms to associate annotations with concept definitions.

2.1.5 Blank Nodes

Among the alternatives to RDF reification and n-ary relations, the only approach that relies on blank nodes to capture provenance is N-Quads. However, blank nodes have to be mentioned not only to emphasize that they cannot be dereferenced, but also because they can be useful in certain scenarios, such as for network discovery tasks, where blank nodes may be useful for collecting statements for subjects that could not be named at the time the task was performed. In fact, such blank nodes can be utilized not only after the subject has been identified, but even during network knowledge discovery. For example, RDF molecules can capture provenance information for two triples that share the same blank node (which cannot be captured with RDF triples or document-level provenance). Note that if an RDF graph has no blank nodes, each triple in the graph constitutes a molecule.

[40] https://purl.org/ontology/network/

2.1.6 Provenance Granularity

The following six levels of provenance granularity can be differentiated from course-grained to fine-grained, depending on the smallest set of represented information for which provenance can be defined:

1. *Dataset-level provenance:* the provenance of Linked (Open) Data datasets. Every statement is globally dereferencable.
2. *RDF document-level provenance:* the provenance of RDF statements stored in the same file.
3. *Graph-level provenance:* statements are made to capture the provenance of named graphs, whose URIs are utilized in quadruples to declare coarse provenance information. It can be used when a set of RDF statements share the same provenance data. In case the provenance data applies to an entire set of RDF triples and not just a subset of them, graph-level provenance is identical to document-level provenance.
4. *Molecule-level provenance:* RDF molecules introduce a granularity level finer than named graphs but coarser than triples, constituting the finest components of lossless RDF graph decomposition for provenance tracking situations when graph-level provenance would result in low recall and triple-level provenance in low precision. Molecule-level provenance provides high precision, because all the RDF documents asserting at least one molecule of a given RDF graph (partially) justify the graph.
5. *Triple-level provenance:* provenance information is provided for RDF triples. This is the most common prove-

Table 7 Some approaches can capture either coarse or fine-grained provenance only

Approach	Provenance Granularity
RDF/XML Source Declaration	Sets of triples, RDF graph
aRDF	Triple
RDF Molecule	Molecule, RDF document
N3Logic	Triple
RDF$^+$	Triple
SPOTL	Triple
RDF*	Triple
Annotated RDFS	Triple, RDF graph, RDF document
PaCE	Triple (application-aware)
Singleton property	Predicate
N-Quads	Triple
Named graphs (incl. GraphSource)	Set of triples, RDF graph, or RDF document
RDF/S graphsets	Triple, set of triples (even if derived from multiple named graphs), RDF graph (covering both explicit and implicit statements)
RDF triple coloring	Set of elements, triple, collection of triples
Nanopublications	Set of triples, RDF graph, or RDF document

nance level for RDF data, because it can represent the provenance of statements, which is adequate for a number of applications. Triple-level provenance offers high recall. For example, two RDF graphs containing a triple with a unique identifier as the object implies that the two subjects are identical (even if they may be named differently), and therefore, all the statements about that subject in these graphs are relevant.

6. *Element-level provenance:* fine-grained provenance that enables to track how individual elements of RDF triples have been derived from other RDF triple elements. Many mechanisms to capture provenance cannot assign provenance to arbitrary statement elements, i.e., subjects, predicates, and objects of RDF triples, only to one of them. Statement-element-level provenance is useful for representing various claims of disputed or uncertain information from diverse sources. Some of these might be contradictory, which can be handled by considering the trustworthiness, reputation, reliability, and quality of the data sources with weight values or preference order. Element-level provenance can also be used in entity resolution.

Table 7 summarizes the RDF reification alternatives from the provenance granularity point of view.

Note that most approaches capture triple- or higher-level provenance, although there are options to capture element-level provenance as well. The singleton property approach captures provenance for predicates, while resource annotation can be used to track how individual triple elements of annotations were derived from triple elements of other annotations.

2.1.7 Scalability

Those approaches that lead to triple bloat are not *scalable*, and hence, they are not suitable for Big Data applications. Some approaches may be scalable at a particular level only, such as at the triple level (see Table 8).

2.2 Knowledge Organization Systems for Provenance

Knowledge organization systems designed for working with RDF data provenance include purpose-built and related controlled vocabularies and ontologies. The next sections will give a brief overview of these vocabularies and ontologies.

2.2.1 Provenance Vocabularies and Ontologies

Various vocabularies and ontologies are available for representing specific types and aspects of provenance information, such as attributes, characteristics, licensing, versioning [79], proof [80], and entailment. These include upper ontologies, which can be used across knowledge domains, domain ontologies that provide provenance terms for specific knowledge domains [81], and provenance-related ontologies, which define terms often captured together with provenance, such as to capture trust and licensing information.

Upper Ontologies for Provenance There is a wide range of domain-agnostic ontologies to represent generic provenance data. The core data model for provenance, PROV,[41] was stan-

[41] https://www.w3.org/TR/prov-dm/

Table 8 Not all reification alternatives are scalable

Approach	Scalable?
RDF/XML Source Declaration	–
aRDF	+
RDF Molecule	Depends on implementation
N3Logic	+
RDF$^+$	–
SPOTL	–
RDF*	–
Annotated RDFS	Depends on implementation
PaCE	–
Singleton property	–
N-Quads	+
Named graphs (incl. GraphSource)	+
RDF/S graphsets	+
RDF triple coloring	+
Nanopublications	+

dardized in 2013 by the W3C [82], serving as the basis for the *Provenance Interchange Ontology (PROV-O)*,[42] the de facto standard provenance ontology [83]. The *Open Provenance Model Ontology (OPMO)*[43] is an OWL ontology for the Open Provenance Model, which extends OPMV by defining more constraints using complex OWL 2 constructors. It was designed to allow provenance information exchange and technology-independent capturing of multi-level provenance, and defines inference rules to identify the validity of provenance inferences. The *Proof Markup Language (PML) Ontology* is an OWL ontology for general-purpose provenance interlingua [80, 84]. Provenir is an upper-level provenance ontology written in OWL DL, which covers core concepts related to information manipulation that can be used across knowledge domains [85]. As mentioned earlier, the PaCE approach utilizes the term `derivesFrom` from this ontology. An extension of Provenir is the *Janus Ontology*, which models the semantic provenance terms that are adequate for representing the domain semantics of workflows [86]. The *Provenance, Authoring and Versioning Ontology (PAV)*[44] defines concepts and properties for describing general, data creation, and data access provenance of web data [87].

Domain Ontologies for Provenance Domain-aware provenance ontologies can be used to represent provenance for specific knowledge domains, e.g., broadcasting, workflows, and scientific processes. The *BBC Provenance Ontology (BBCPROV)*[45] supports data management and auditing tasks. It is suitable for defining different types of named graphs used in quadstores, and associate them with metadata to manage, validate and expose data to services of the British Broadcasting Corporation. The *Ontology for Provenance and Plans (P-Plan)*[46] is an extension of the PROV-O ontology for the representation of how-provenance for plans used to execute scientific processes. The *Wfprov Ontology (WFPROV)*[47] can express provenance information about the execution of a workflow. The *Open Provenance Model for Workflows (OPMW)*[48] is an ontology for describing workflow traces and their templates based on the Open Provenance Model. The *Open proVenance Ontology (OvO)* was design to support scientific experiments [88]. *PREMIS*[49] is the OWL implementation of the U.S. Government's provenance vocabulary of the same name. It supports long-term preservation, with a focus on the provenance of archived digital objects, such as files, bitstreams, and aggregations, rather than the provenance of descriptive metadata [89].

[42] http://www.w3.org/ns/prov-o

[43] https://github.com/KRAETS/ccerschema/blob/master/KRAETS/ccerschema/core/third_party/provenance/opmo-20101012.owl

[44] http://purl.org/pav/

[45] https://www.bbc.co.uk/ontologies/provenance/1.9.ttl

[46] http://vocab.linkeddata.es/p-plan/p-plan.owl

[47] http://purl.org/wf4ever/wfprov

[48] http://www.opmw.org/model/OPMW/

[49] http://premisontologypublic.pbworks.com/w/file/fetch/58521655/premis2.2_v0.1.owl

2.2.2 Provenance Vocabularies

The *Provenance Vocabulary (PRV)*[50] is an OWL 2 vocabulary designed for tracking information manipulation [90]. It is a Web data-specific specialization of the PROV Ontology, and defines core concepts for tracking data creation and access, and concepts of data transfer and information authentication in a taxonomical structure. The *Open Provenance Model Vocabulary (OPMV)*,[51] a lightweight vocabulary designed to assert *Open Provenance Model (OPM)*[52] concepts, and the *Open Provenance Model OWL Ontology (OPMO)*, which extends OPM to support inferencing [91]. The *Vocabulary of Interlinked Datasets (VoID)*[53] is suitable for providing generic dataset-level provenance information. The provenance extension of VoID, called *VoIDP*, can be used to answer queries, such as how data was derived, who carried out the transformation, and what processes have been used for the transformations [92,93]. The OAI-ORE Terms Vocabulary is the provenance vocabulary of the Open Archives Initiative for the description and exchange of aggregations of web resources [94].

The *Semantic Web Publishing Vocabulary (SWP)*[54] is an RDFS vocabulary for capturing information provision and assuring the origin of information with digital signatures. *Web Annotation Vocabulary*,[55] used by the Web Annotation Data Model[56] to annotate web resources in JSON-LD. The *POWDER Vocabulary (WDR)*[57] of the Protocol for Web Description Resource[58] can be used to describe a group of resources, such as by using user-defined tags associated with a semantically explicit description.

2.2.3 Provenance-Related Ontologies

Dublin Core (DC),[59] standardized in ISO 15836-1:2017,[60] is a set of 25 elements that can be broadly classified as provenance-related, including one generic term, namely, `provenance`, and terms of three specific provenance categories: terms that capture who affected a change (`contributor`, `creator`, `publisher`, `rightsHolder`), terms to answer questions about when a change was affected (`available`, `created`, `date`, `dateAccepted`, `dateCopyrighted`, `dateSubmitted`, `issued`, `modified`, `valid`), and terms that can be used to describe how a change was affected (`isVersionOf`, `hasVersion`, `isFormatOf`, `hasFormat`, `license`, `references`, `isReferencedBy`, `replaces`, `isReplacedBy`, `rights`, `source`). The DC Terms can partially be mapped to PROV-O terms [95]. *Creative Commons*[61] is an RDFS ontology for describing licensing information, some of which are provenance-related. The *Changeset Vocabulary*[62] can be used to store the changes between two versions of a resource description. The *Web of Trust Ontology (WOT)*[63] defines terms for describing how the validity of data items has been assured through encryption or digital signature. In particular, WOT captures provenance data, such as the timestamp and key of digital signatures. The *Trust Assertion Ontology (TAO)*[64] is a lightweight ontology to describe subjective trust values of users. Ontologies designed to capture metadata beyond (or not specifically for) provenance, such as temporal constraints, may also be suitable to capture provenance (e.g., *4DFluents* [96], *NDFluents* [97]).

3 Provenance-Aware RDF Data Management

The graph model that powers graph databases is fundamentally a match for the core of provenance [98]. The importance of provenance attributes caught the attention of graph database vendors, resulted in proprietary implementations for storing various properties for RDF triples and quadruples [99]. The efficient implementation of provenance-enabled queries resulted in novel indexing techniques for RDF provenance [100] and approaches such as *provenance polynomials* [101] and *adaptive RDF query processing* [102].

3.1 Provenance-Aware RDF Data in Graph Databases

AllegroGraph[65] is an industry-leading graph database famous for not only its high scalability over millions of quads, but also for its support for additional fields at the triple level, making it possible to define permissions, trust, and provenance data for source tracking, quality evaluation, and access control. AllegroGraph supports a format called *Extended N-Quad*, or *NQX* for short, which extends the standard N-Quads

[50] http://purl.org/net/provenance/ns
[51] http://purl.org/net/opmv/ns
[52] https://openprovenance.org
[53] https://www.w3.org/TR/void/
[54] https://www.w3.org/2004/03/trix/swp-2/
[55] https://www.w3.org/TR/annotation-vocab/
[56] https://www.w3.org/TR/annotation-model/
[57] http://www.w3.org/2007/05/powder#
[58] https://www.w3.org/TR/powder-dr/
[59] http://purl.org/dc/terms/dcterms.ttl
[60] https://www.iso.org/standard/71339.html
[61] https://creativecommons.org/schema.rdf
[62] http://vocab.org/changeset/schema-20090518.rdf
[63] http://xmlns.com/wot/0.1/index.rdf
[64] http://vocab.deri.ie/tao.ttl
[65] https://franz.com/agraph/allegrograph/

format to allow the specification of optional attributes for each triple or quad in JSON format.[66] NQX allows an arbitrary number of attributes and an arbitrary number of attribute values, with a maximum attribute size limited only by the amount of available virtual memory and address space (theoretically up to approximately 1TB). Because the permissible characters in attribute names are restricted to a composition of lower ASCII characters, including letters, digits, dashes, and underscores, and URIs may contain characters beyond this character set, this implementation would only be an option if the provenance attributes would be declared on quads, i.e., the graph identifier would be defined as part of the quadruples, rather than an attribute value. However, attributes can be defined only when adding the triples/quads to the repository, i.e., their associated attributes cannot be changed or removed afterwards.

Neo4j[67] is a graph database, which employs a property graph model. This model allows the definition of properties for both roles and relationships, and labels to assign roles or types to nodes. These features are suitable for, among others, storing provenance data, as evidenced by implementations such as the CAPS framework [103] and MITRE's provenance management software, PLUS.[68]

OpenLink Virtuoso[69] supports additional metadata to be stored with RDF triples, which can be used for representing provenance data [104]. However, how to add provenance data to triples in Virtuoso is not trivial, because it requires a kind of mechanism that extends the standard SPARQL query syntax [105].

D2R Server[70] can be used to expose data from relational databases as RDF. It implements PROV-O and supports provenance information, along with other metadata, to be attached to every RDF document and web page published with it.

3.2 Provenance-Aware LOD Datasets

Provenance data can be used in LOD datasets to facilitate information fusion, thereby avoiding technology-specific analytics that might be biased toward certain data sources and eliminating the need for manually pulling information together [106]. Data inferred by software agents should be distinguished from data explicitly provided by data publishers, because they differ in terms of trustworthiness [107]. Without sufficient transparency for Linked Data sources and transformations, government agencies and scientists cannot trust third-party LOD datasets [82]. The data interlinking mechanism used by LOD datasets can be utilized for coarse provenance information in the form of data associations, but these do not cover data transformations [108]. The `owl:sameAs` predicate, which is widely deployed among LOD datasets [109], can result in the confusion of provenance and ground truth [110]. The provenance information provided by named graphs indicate the current location of data, or the data source described by provenance graphs [111], but does not hold information about the behavior of processes that interact with Linked Data, which can be captured using additional syntax and semantics only [112]. The aforementioned VoID vocabulary can be used to provide dataset-level provenance for LOD datasets. However, a complete provenance chain may be required for some applications, allowing every single statement to be the subject of annotations and links [113]. Provenance descriptors can be published as Linked Data in two ways: either a link represents an entity and links directly to provenance properties, or a provenance property links the URI to the starting point of a provenance descriptor [114]. These links allow mechanisms to be implemented for automatically defining provenance information during data integration [115].

Since datasets enriched with data provenance may have duplicate provenance values, techniques have been proposed to eliminate these, thereby optimizing the storage of RDF data provenance [116]. Hybrid storage models exist for Linked Data to exploit recurring graph patterns and graph partitioning, which enable complex cloud computations on Big RDF Data, making the provenance-aware management of Linked Data efficient and scalable [117]. LOD datasets enriched with provenance data can contain domain-agnostic provenance graphs or domain-aware provenance graphs, the latter of which can answer far more specific queries [86]. Tracking data provenance may require both generic and domain-specific provenance data to support future reuse via querying, and provenance traces from diverse resources often require preservation and interconnection to support future aggregation and comparison [118]. Provenance-aware Linked Data querying consists of a workload query and a provenance query [119], which can be executed with various strategies, such as the following [120]:

- Post-filtering: the independent execution of the workload and provenance query;
- Query rewriting: the execution of the provenance query precedes the workload query, making it possible to utilize context values returned by the provenance query to filter out those tuples that do not conform to the provenance results;
- Full materialization: the provenance query is executed on the entire dataset or any relevant subset of it, and

[66] https://franz.com/agraph/support/documentation/6.1.6/triple-attributes.html

[67] https://neo4j.com

[68] https://github.com/plus-provenance/plus

[69] https://virtuoso.openlinksw.com

[70] http://d2rq.org/d2r-server

materializes all tuples whose context values satisfy the provenance query;
- Pre-filtering: a provenance index is located for each context value and identifier of those tuples that belong to the context;
- Adaptive partial materialization: introduced a trade-off between the performance of the provenance query and that of the workload query.

4 Querying RDF Provenance

Querying RDF data provenance is not trivial because of nonstandard provenance representations that capture static provenance data and the lack of support for version control for RDF. Therefore, the following approaches have been introduced.

Damásio et al. developed an approach for querying provenance information for LOD obtained from SPARQL endpoints [121]. This approach translates SPARQL into annotated relational algebra, in which the annotated relations have values from the most general m-semiring.

Halpin and Cheney proposed a technique to facilitate querying dynamic changes of RDF graphs using SPARQL [122]. This technique is compatible with W3C's PROV, and allows the definition of provenance information by reinterpreting SPARQL updates [123].

Avgoustaki et al. proposed a provenance model that borrows properties from the how and the where provenance models, allowing to capture triple-level and attribute-level provenance of data added to datasets via SPARQL `INSERT` updates [124].

Another algebraic structure suitable for capturing the provenance of SPARQL queries is the evaluation of SPARQL algebra queries on RDF graphs annotated with elements of spm-semirings [125]. These extend semirings with an operator to capture the semantics of non-monotone SPARQL operators.

5 Software Tools for Manipulating RDF Provenance

Utilizing data provenance in RDF-based applications allows data quality assessment, find similar or related resources, and makes Linked Open Data concept interlinking more efficient [126]. In web applications, provenance can be used not only to provide information about the trustworthiness of the data, but also to support project collaboration, identify errors in data sources, extend insights to other applications [127], describe and discover web services [128], comment tracking [129], improve interoperability, and so on.

The *Provenance Tool Suite*[71] includes provenance software tools including *Prov Python (ProvPy)*,[72] a Python library supporting PROV-DM data import and export as PROV-XML,[73] and PROV-JSON;[74] *ProvToolbox*,[75] a Java library to create Java representations of PROV-DM and convert them to PROV-O, PROV-XML, PROV-N, and PROV-JSON; and *ProvJS*, a JavaScript utility for indexing and searching PROV-JSON objects within JavaScript objects. These packages are employed by three services:[76] *ProvStore*,[77] a free provenance-aware repository hosting PROV provenance documents; *ProvTranslator*,[78] which translates PROV documents across different PROV representations; and *ProvValidator*[79] for validating PROV documents. These services can be used via REST API or a browser interface.

Beyond Provenance Tool Suite's ProvValidator, there are other tools to check provenance data, such as *prov-check*[80] and *prov-constraints*.[81]

One of the six web applications of the *eagle-I* software suite,[82] a resource discovery tool for translational science research, is an RDF repository that can be used to store resource and provenance metadata as RDF triples. The Linked Open Data output can be exported via SPARQL endpoints, flat RDF files, or published URI lists.

Provenance Explorer was designed to provide a customizable visualization of the provenance trail associated with scientific discovery processes by utilizing both explicit and implicit RDF relationships [130]. *LabelFlow* is a tool to manipulate the workflow provenance of scientific data in RDF [131]. It enables semi-automated provenance annotation and can handle PROV-O- and WFPROV-compliant provenance traces. *Taverna*,[83] Apache's open source workflow management system allows the export of workflow run provenance as PROV-O annotations in RDF through its PROV plugin. This plugin can export the workflow execution (output and intermediate values) and the provenance trace as a PROV-O RDF graph, which can be queried using SPARQL

[71] https://www.software.ac.uk/who-do-we-work/provenance-tool-suite

[72] https://pypi.python.org/pypi/prov

[73] https://www.w3.org/TR/prov-xml/

[74] https://www.w3.org/Submission/prov-json/

[75] https://github.com/lucmoreau/ProvToolbox/

[76] https://openprovenance.org

[77] https://openprovenance.org/store/

[78] https://openprovenance.org/services/view/translator

[79] https://openprovenance.org/services/view/validator

[80] https://github.com/pgroth/prov-check

[81] https://github.com/jamescheney/prov-constraints

[82] https://www.eagle-i.net/get-involved/for-developers/

[83] https://github.com/apache/incubator-taverna-engine/tree/master/taverna-prov

and processed with other PROV tools, such as the PROV Toolbox and the *Provenance Extractor*,[84] a command line tool to extract provenance information form Taverna provenance databases.

The *Core Provenance Library (CPL)* is a portable provenance library to be incorporated in a variety of software tools to collect and integrate provenance data [132]. It can work with both relational and graph databases via its ODBC and RDF/SPARQL drivers. *PROVoKing*[85] is a Java library for the PROV standard. By utilizing Apache Jena, it reads PROV data from a Turtle, RDF/XML, N3, or N-Triple file, a URI, or a Jena Model produced from SPARQL query answers. PROVoKing converts a PROV document to an RDF triple stream in the memory. It prefers binary PROV-O relationships over qualified PROV-O relations. Provenance data can be exported to Turtle. The *prov-api*[86] is a Java API to create and manipulate provenance graphs using core PROV terminology. *Tupelo* is an open source semantic content management framework, which can manage a range of metadata, including provenance [133]. It can be used to develop applications that list metadata associated with entities and visualize provenance graphs.

DeFacto[87] is an application that implements PROV-O for deep fact validation, i.e., finding confirming sources for statements on the Web [134]. *ProvRPQ* is a tool for provenance-aware regular path queries (RPQs), which are used to express navigations over RDF graphs [135]. This interactive querying tool can clearly justify how paths conforming to RPQs can be navigated from source to target resources in RDF graphs by expanding conventional answers and introducing witness resources. The *Hedgehog RDF Publisher*[88] is a system for publishing datasets in RDF. Its scripts pull data from a source and expose the data as RDF triples. It can automatically add metadata about the data, including provenance. *Pubby*[89] is a Linked Data frontend for SPARQL endpoints with a metadata extension that provides provenance information.

RDF provenance tools are not limited to domain-agnostic software libraries and APIs, as some of the tools are domain-specific. A prime example is *Photostuff*, a platform-independent, open source image annotation tool that allows image and image region annotation with arbitrary ontology terms [36]. It exploits and provides support for provenance management, and employs annotations such as submitter name and email to track image provenance. In Photostuff, provenance data can be browsed directly and is also used to enrich user experience.

6 Provenance Applications

Applications using provenance can be found in many domains, especially where evidence or context-awareness are important, such as in cybersecurity and the medical domain, as briefly described in this section.

GraphSource has been used for cyber-situational awareness to detect inconsistencies and changes in network topologies and paths, which may be contributing to cybersecurity incidents [136]. A digital forensics system called *ParFor* used the named graphs approach to provide context of each assertion [137]. This enabled files and users to be correlated across multiple devices, thereby offloading much manual effort of human investigators. Motivated by controlling privacy concerns and security access control to RDF triples, Lopes et al. [138] used annotated RDF to manage permissions and query in a domain-specific way. Inspired by named graphs and annotated RDF, an ambient intelligence system is able to detect a violation if a person is found to be in two locations at once. This includes an indicator of certainty when time frames overlap [139].

PaCE was motivated by the Biomedical Knowledge Repository (BKR) [55]. The PaCE approach in BKR allows provenance tracking of two different scientific articles where both stated that Ibuprofen affects inflammitory cells. From this corroboration of evidence-based provenance, confidence values can be inferred. Nanopublications was used to suggest treatment with backup evidence, traceable back to literature including the population on which treatment was studied [140].

7 Performance Comparison of RDF Data Provenance Approaches

The syntactic differences between the presented approaches and techniques are not always accompanied by semantic differences. For example, the context identifier for RDF statements is written differently across quadruple-based and named graph-based approaches, yet it represents the same type of provenance information, making it possible to convert provenance-aware RDF data between them without losing semantics, as long as the datatype and value range are not stricter for one than the other (e.g., N-Quads and named graphs).

Comparing the performance of RDF data provenance approaches is not trivial because of the variety of knowledge domains and provenance representations, the diversity

[84] http://www.ifs.tuwien.ac.at/dp/process/projects/provenanceExtractor.html

[85] https://sites.google.com/site/provokinglibrary/

[86] https://github.com/dcorsar/prov-api/

[87] http://defacto.aksw.org

[88] https://github.com/ads04r/Hedgehog

[89] http://wifo5-03.informatik.uni-mannheim.de/pubby/

Table 9 Types of metrics considered for quantitative comparison, with their corresponding measurement and impact

Metric	Measurement	Impact
File size	Linux command line	Storage
Number of triples	Stardog add command	Querying, reasoning
Number of quads	Stardog add command	Querying, compatibility
Number of graphs	SPARQL `SELECT DISTINCT` graphs	Querying, reasoning
Number of distinct IRIs	SPARQL `SELECT DISTINCT` subjects filtered for IRI	Graph edge count
Average IRI length	SPARQL `SELECT` average length using distinct filters for IRI	Readability
Share of provenance statements	DataSetGS SPARQL `SELECT ALL` triples from PROVENANCE graph; DatasetSP SPARQL `SELECT ALL` triples with predicate `rdf:singletonPropertyOf` and `prov:wasDerivedFrom`	Contextualization

of data formats, the variable support from RDF data management tools, the availability of published datasets using the various RDF data provenance approaches, and the compatibility of software tools for manipulating and converting between the different mechanisms for capturing RDF data provenance.

In addition, the captured semantics often cannot be expressed quantitatively, and some of the vocabulary or ontology terms used as part of a provenance mechanism do not have an explicitly defined meaning. Therefore, even if the same case is represented using different approaches (with the same intended semantics), the overhead caused by the different requirements and the storage and processing of extra (usually meaningless) properties and/or relationships that would not be defined at all without the need of provenance should be judged on a case-by-case basis.

The following sections provide quantitative comparisons of some of the state-of-the-art approaches for capturing RDF provenance from the querying performance point of view.

7.1 Experiments

Our experiments to compare RDF data provenance approaches used multiple formalisms, datasets, and software tools. For the knowledge domain of communications networks, earlier we introduced a quad-based RDF provenance capturing approach [40], *GraphSource*, and developed a provenance-aware LOD dataset, *ISPNet* [106, 111], both of which have been used in the experiments. To convert provenance-aware RDF datasets based on RDF quadruples to datasets that utilize the singleton property to capture provenance, we used the software tool *RDF Contextualizer*[90] designed by Nguyen and Sheth [141]. For RDF data storage and management, we selected *Stardog*,[91] which supports, among other features, quad-based representation. Using Stardog and RDF Contextualizer, the following three datasets have been created:

- *DatasetGS* (GraphSource-based ISPNet dataset)
- *DatasetSP* (singleton property-based ISPNet dataset)
- *DatasetNQ* (N-Quad-based ISPNet dataset)

For the purposes of this section, we focus exclusively on DatasetGS and DatasetSP. This is because Stardog treated DatasetNQ the same way as DatasetGS, meaning that it automatically created named graphs from N-Quads, which would yield identical metrics, queries, and query plan results for the two.

Table 9 shows the metrics used to provide a simple quantitative comparison. The technique for measurement is provided, as well as the most important factors they affect.

These general metrics are suitable for comparing provenance-aware datasets that use different formalisms to capture provenance.

The following types of domain-independent provenance queries were formed to provide a simple quantitative comparison:

- Query 1: select all provenance statements
- Query 2: select all triples for a given data source
- Query 3: select all data sources for a given subject
- Query 4: select all triples for a specific predicate ordered by time
- Query 5: select all triples derived from a specific location at specific time

Each provenance query was written in SPARQL and executed in Stardog. After query execution, the Stardog query plan was retrieved, which provided insight into the complexity of the queries, as the complexity and length of a query plan is generally proportional to the query execution time.[92] In the following section, the results are presented for

[90] https://archive.org/services/purl/rdf-contextualizer

[91] https://www.stardog.com

[92] Clearly differentiating between query execution times of various provenance capturing approaches would require datasets with millions of provenance-aware triples, but the approaches described in the literature have not yet been globally deployed to facilitate this.

Table 10 Metrics-based comparison between implementations of GraphSource (DatasetGS) and singleton property (DatasetSP)

Metric	DatasetGS	DatasetSP
File size [in bytes]	93,647	734,781
Number of triples	N/A	6261
Number of quads	2182	N/A
Number of graphs	49	N/A
Number of distinct IRIs	368	2454
Average IRI length	49.72	56.38
Share of provenance statements	10%	66%

Query 5 (both time and location). This query was selected because for many knowledge domains, including communication networks, context-awareness and context-specific reasoning will likely occur within specific time periods at specific geographic locations (or bounds).

7.2 Results

This section provides the results of a complex query, Query 5, allowing a quantitative comparison between the GraphSource and the singleton property approach based on metrics, queries, and query plans.

Table 10 presents the results of a metrics-based comparison between the DatasetGS and DatasetSP datasets (using GraphSource and the singleton property, respectively).

These metrics and their impact (discussed earlier in Sect. 7.1) show that GraphSource implementations outperform that of the singleton property approach in terms of file size, number of distinct IRIs, average IRI length and knowledge statement-provenance statement ratio. This indicates that, generally speaking, GraphSource is superior from several points of view, including impact on storage, queries, graph edge count, human-readability, and contextualization.

7.3 Queries

Figure 2 presents a comparison of SPARQL queries between the DatasetGS (GraphSource-based) and DatasetSP (singleton property-based).

For DatasetGS, the SPARQL query is 14 lines (see the left-hand side of Fig. 2). The query uses the PROVENANCE named graph to get the context name *g*, i.e., the named graph identifier (see line 6). From this context, it selects the `net:importTime` and `net:importHost` (location) (see lines 7 and 8). Then, on line 10, from within the context *g*, it selects the desired predicate, which is `net:hasInterface` in this example (see line 11). For

```
1  prefix prov: <http://www.w3.org/ns/prov#>
2  prefix net: <http://purl.org/ontology/network/>
3  prefix ispnet: <http://purl.org/dataset/ispnet/base/>
4  SELECT ?h ?t ?g ?s ?o
5  WHERE {
6   graph <http://purl.org/dataset/ispnet/base/PROVENANCE> {
7     ?g net:importTime ?t .
8     ?g net:importHost ?h .
9   }
10  graph ?g {
11    ?s net:hasInterface ?o .
12  }
13 }
14 order by ?h ?t ?g ?s ?o
```

```
1  prefix prov: <http://www.w3.org/ns/prov#>
2  prefix net: <http://purl.org/ontology/network/>
3  prefix ispnet: <http://purl.org/dataset/ispnet/base/>
4  prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
5  SELECT ?h ?t ?g ?s ?o
6  WHERE
7  {
8    ?spt rdf:singletonPropertyOf net:importTime .
9    ?spt prov:wasDerivedFrom <http://purl.org/dataset/ispnet/base/PROVENANCE> .
10   ?g ?spt ?t .
11   ?sph rdf:singletonPropertyOf net:importHost .
12   ?sph prov:wasDerivedFrom <http://purl.org/dataset/ispnet/base/PROVENANCE> .
13   ?g ?sph ?h .
14   ?spo rdf:singletonPropertyOf net:hasInterface .
15   ?spo prov:wasDerivedFrom ?g .
16   ?s ?spo ?o .
17 }
18 order by ?h ?t ?g ?s ?o
```

Fig. 2 Query length and complexity comparison based on Query 5 for GraphSource (left) versus singleton property (right)

```
1 Projection(?h, ?t, ?g, ?s, ?o) [#268]
2 `— OrderBy(ASC(?h), ASC(?t), ASC(?g), ASC(?s), ASC(?o)) [#268]
3   `— HashJoin(?g) [#268]
4     +— Scan[PSOC](?s, net:hasInterface, ?o){?g} [#268]
5     `— MergeJoin(?g) [#1]
6       +— Scan[PSO](?g, net:importTime, ?t){ispnet:PROVENANCE} [#44]
7       `— Scan[PSO](?g, net:importHost, ?h){ispnet:PROVENANCE} [#44]
```

```
1  Projection(?h, ?t, ?g, ?s, ?o) [#1]
2  `— OrderBy(ASC(?h), ASC(?t), ASC(?g), ASC(?s), ASC(?o)) [#1]
3    `— MergeJoin(?spt) [#1]
4      +— Sort(?spt) [#1]
5      |  `— MergeJoin(?spo) [#1]
6      |    +— Sort(?spo) [#1]
7      |    |  `— MergeJoin(?g) [#1]
8      |    |    +— Sort(?g) [#1]
9      |    |    |  `— MergeJoin(?spo) [#1]
10     |    |    |    +— Scan[POSC](?spo, rdf:singletonPropertyOf, net:hasInterface) [#268]
11     |    |    |    `— Scan[PSOC](?spo, prov:wasDerivedFrom, ?g) [#2.1K]
12     |    |    `— HashJoin(?sph) [#1]
13     |    |      +— MergeJoin(?g) [#1]
14     |    |      |  +— Scan[SPOC](?g, ?sph, ?h) [#6.3K]
15     |    |      |  `— Scan[SPOC](?g, ?spt, ?t) [#6.3K]
16     |    |      `— MergeJoin(?sph) [#1]
17     |    |        +— Scan[POSC](?sph, rdf:singletonPropertyOf, net:importHost) [#44]
18     |    |        `— Scan[POSC](?sph, prov:wasDerivedFrom, ispnet:PROVENANCE) [#225]
19     |    `— Scan[POSC](?s, ?spo, ?o) [#6.3K]
20     `— MergeJoin(?spt) [#1]
21       +— Scan[POSC](?spt, rdf:singletonPropertyOf, net:importTime) [#44]
22       `— Scan[POSC](?spt, prov:wasDerivedFrom, ispnet:PROVENANCE) [#225]
```

Fig. 3 Query structure analysis to identify pipeline breakers in the case of GraphSource (left) and singleton property (right)

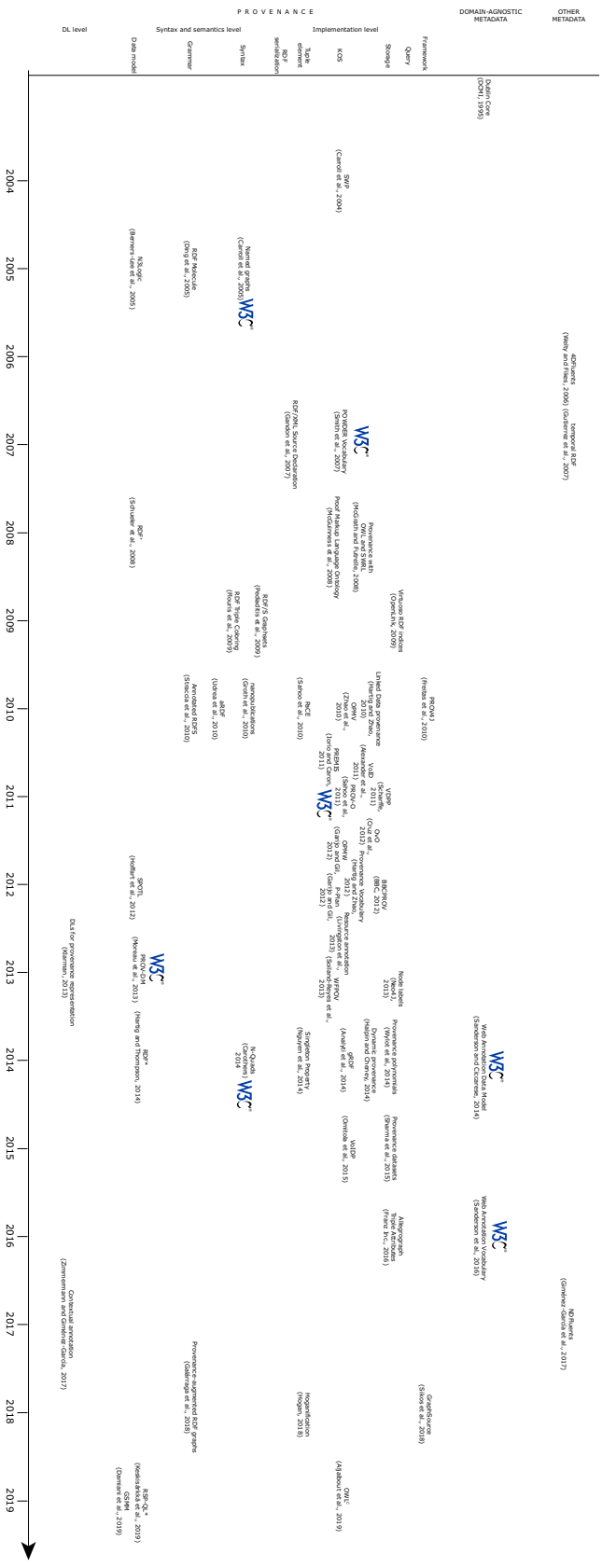


Fig. 4 Timeline of RDF Reification Alternatives. The W3C logo indicates standardization (technical specifications) by the World Wide Web Consortium

convenience, the results are ordered by location and time on line 14.

In the case of DatasetSP, the SPARQL query is 18 lines (see the right-hand side of Fig. 2). The query first uses the `rdf:singletonPropertyOf` to get the `net:importTime` from the `PROVENANCE` named graph (see lines 8 and 9), and get context *g* (see line 13). The query then repeats the process to get the `net:importHost` (see lines 11–13). Next, now having the correct context *g*, `rdf:singletonPropertyOf` is needed for `net:hasInterface` (see line 14) and ensure this predicate comes from the same context *g* using `prov:wasDerivedFrom` (see line 15). Finally, on line 16, it can retrieve the subject and the object. Similarly to the previous query, the results are ordered by time and location (see line 18).

Figure 3 presents a comparison of the query plan between the GraphSource-based DatasetGS and the singleton property-based DatasetSP.

The main considerations regarding Stardog query plans include cardinality estimations, bottom-up semantics, and pipeline breakers, the latter of which have a significant impact on query performance, being those operators that require intermediate results to be evaluated first before continuing query execution [142]. These include `HashJoin`, `Sort`, and `GroupBy`.

For DatasetGS, the SPARQL query plan is 7 lines (see the left-hand side of Fig. 3). The two scans are evaluated first on lines 6 and 7 (bottom-up semantics) with a cardinality estimation of 44. This follows with an efficient `MergeJoin` on line 5, `Scan` on line 4 with a cardinality of 268, followed by a single pipeline breaker, `HashJoin`, on line 2.

In contrast, the SPARQL query plan of DatasetSP is three times longer (the GraphSource query plan is 7 lines, in contrast to the 22 lines of the same query implemented using the singleton property approach, as seen in Fig. 3). The innermost scans are evaluated first (lines 10–11, 14–15, and 17–18). The cardinality estimations are much higher compared to DatasetGS, for example 6300 on line 14. While there are a number of efficient `MergeJoins`, there are also four pipeline breakers: a `HashJoin` (on line 12) and three `Sorts` (on lines 4, 6, and 8). Considering these pipeline breakers, it is not surprising that DatasetGS produced lower execution times compared to DatasetSP, and this difference would increase exponentially with the size of the datasets.

8 Conclusions

Developing a single mechanism for integrating RDF statements with provenance data that is formally grounded and is not only implementable using standard languages, but also scalable, has long been in the center of attention of provenance research. Several alternatives have been proposed

to the highly criticized RDF reification (as summarized in Fig. 4), all with different representation prerequisites, provenance granularity and precision, and reasoning potential.

These approaches differ in terms of the mechanism they employ to capture provenance, such as by extending the RDF data model, using a class instance, defining an instantiated property, or utilizing a graph that contains the relational assertions. Because these approaches capture different facets of provenance, the implementation choice depends on the applications. Nevertheless, the comprehensive comparisons presented in this paper help implementers find the most suitable solution for their projects.

At a higher level of abstraction, there is a variety of knowledge organization systems that can be utilized in capturing provenance-aware RDF statements, including purpose-built controlled vocabularies and ontologies, and ontologies designed for general or other types of metadata. Storing provenance-aware RDF statements requires solutions that go beyond the capabilities of conventional triplestores, and either encapsulate metadata with the triples, or store more than three columns per statement to capture provenance (quadstores, graph databases). This paper enumerated these solutions, and reviewed how to run queries on provenance-aware RDF statements not only on a single, but also on multiple datasets (federated queries), including update operations. Furthermore, software tools for manipulating RDF data provenance have also been discussed, noting that while PROV and OPM can be implemented in software tools with RDF support, not all tools that implement PROV or OPM can actually handle RDF files. A variety of scenarios require a trust mechanism that can be supported by capturing data provenance. The research interest in RDF data provenance indicates the importance of this field, for intelligent systems implementing Semantic Web standards need provenance manipulating capabilities to be viable, particularly in systems where RDF triples are derived from diverse sources, are generated and processed on the fly, or modified via update queries.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Mojžiš J, Laclavík M (2013) SRelation: fast RDF graph traversal. In: Klinov P, Mouromtsev D (eds) Knowledge engineering and the Semantic Web. Springer, Heidelberg, pp 69–82. https://doi.org/10.1007/978-3-642-41360-5_6
2. Sikos LF, Choo KKR (eds) (2020) Data science in cybersecurity and cyberthreat intelligence. Springer, Cham. https://doi.org/10.1007/978-3-030-38788-4
3. Sikos LF (2015) Mastering structured data on the Semantic Web. Apress, New York. https://doi.org/10.1007/978-1-4842-1049-9
4. Sikos LF (2019) Knowledge representation to support partially automated honeypot analysis based on Wireshark packet capture files. In: Czarnowski I, Howlett RJ, Jain LC (eds) Intelligent decision technologies 2019. Springer, Singapore, pp 345–351. https://doi.org/10.1007/978-981-13-8311-3_30
5. Sikos LF (2020) Packet analysis for network forensics: a comprehensive survey. Forensic Sci Int Digit Investig 32C:200892. https://doi.org/10.1016/j.fsidi.2019.200892
6. Moreau L (2006) Usage of 'provenance': a Tower of Babel–towards a concept map. Life Cycle Seminar, Mountain View
7. Pérez B, Rubio J, Sáenz-Adán C (2018) A systematic review of provenance systems. Knowl Inf Syst. https://doi.org/10.1007/s10115-018-1164-3
8. McGlothlin JP, Khan L (2010) Efficient RDF data management including provenance and uncertainty. In: Proceedings of the Fourteenth International Database Engineering and Applications Symposium. ACM, New York, pp 193–198. https://doi.org/10.1145/1866480.1866508
9. Moreau L (2010) The foundations for provenance on the Web. J Found Trends Web Sci 2(2–3):99–241. https://doi.org/10.1561/1800000010
10. Garae J, Ko RKL (2017) Visualization and data provenance trends in decision support for cybersecurity. In: Carrascosa IP, Kalutarage HK, Huang Y (eds) Data analytics and decision support for cybersecurity. Springer, Cham, pp 243–270. https://doi.org/10.1007/978-3-319-59439-2_9
11. Springer, Cham. (2018) AI in cybersecurity. https://doi.org/10.1007/978-3-319-98842-9
12. Sikos LF (2018) Handling uncertainty and vagueness in network knowledge representation for cyberthreat intelligence. In: 2018 IEEE International Conference on Fuzzy Systems. IEEE, Piscataway. https://doi.org/10.1109/FUZZ-IEEE.2018.8491686
13. Sikos LF, Stumptner M, Mayer W, Howard C, Voigt S, Philp D, (2018) Summarizing network information for cyber-situational awareness via cyber-knowledge integration. AOC 2018 Convention. Adelaide, Australia, 28–30 May 2018
14. Pandey M, Pandey R (2014) Analysis of provenance data stack for OWL ontology relevance. In: Singh Y, Sehgal V, Nitin, Ghrera SP (eds) Proceedings of the 2014 International Conference on Parallel, Distributed and Grid Computing. IEEE, Washington, pp 365–369. https://doi.org/10.1109/PDGC.2014.7030772
15. Pandey M, Pandey R (2015) Provenance constraints and attributes definition in OWL ontology to support machine learning. In: Guerrero J (ed) Proceedings of the 2015 International Conference on Computational Intelligence and Communication Networks. IEEE, Washington, pp 1408–1414. https://doi.org/10.1109/CICN.2015.334
16. Dellal I, Jean S, Hadjali A, Chardin B, Baron M (2019) Query answering over uncertain RDF knowledge bases: explain and

obviate unsuccessful query results. Knowl Inf Syst 61(3):1633–1665. https://doi.org/10.1007/s10115-019-01332-7
17. Fu G, Bolton E, Queralt N, Furlong LI, Nguyen V, Sheth A, Bodenreider O, Dumontier M (2015) Exposing provenance metadata using different RDF models. In: Malone J, Stevens R, Forsberg K, Splendiani A (eds) Proceedings of the 8th International Conference on Semantic Web Applications and Tools for Life Sciences. RWTH Aachen University, Aachen, pp 167–176
18. Suchanek FM, Lajus J, Boschin A, Weikum G (2019) Knowledge representation and rule mining in entity-centric knowledge bases. In: Krötzsch M, Stepanova D (eds) Reasoning Web. Explainable artificial intelligence. Springer, Cham, pp 110–152. https://doi.org/10.1007/978-3-030-31423-1_4
19. Moreau L (2010a) Foundations and trends: the foundations for provenance on the Web. Now Publishers, Hanover
20. Lopes N, Zimmermann A, Hogan A, Lukácsy G, Polleres A, Straccia U, Decker S (2010) RDF needs annotations. In: RDF next steps, Stanford, Palo Alto, CA, USA, June 26–27, 2010
21. Zhao J, Bizer C, Gil Y, Missier P, Sahoo S (2010) Provenance requirements for the next version of RDF. In: RDF Next Steps, Stanford, Palo Alto, CA, USA, June 26–27, 2010
22. Moreau L, Groth P, Herman I, Hawke S (2013) Provenance WG Wiki. https://www.w3.org/2011/prov/wiki/Main_Page. Accessed 29 March 2020
23. Li X, Lebo T, McGuinness DL (2010) Provenance-based strategies to develop trust in Semantic Web applications. Provenance and annotation of data and processes 182–197: https://doi.org/10.1007/978-3-642-17819-1_21
24. Chen L, Jiao Z, Cox SJ (2006) On the use of semantic annotations for supporting provenance in grids. In: Nagel WE, Walter WV, Lehner W (eds) Euro-Par 2006 Parallel Processing. Springer, Heidelberg, pp 371–380. https://doi.org/10.1007/11823285_38
25. Chen L, Yang X, Tao F (2006) A Semantic Web service based approach for augmented provenance. In: Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence. IEEE Computer Society, Los Alamitos, CA, USA, pp 594–600. https://doi.org/10.1109/WI.2006.25
26. Chen L, Jiao Z (2006) Supporting provenance in service-oriented computing using the Semantic Web technologies. IEEE Intell Inform Bull 7(1):4–11
27. Sahoo SS, Barga RS, Goldstein J, Sheth AP (2008) Provenance algebra and materialized view-based provenance management. Technical Report 76523/tr-2008-170
28. Miles S, Wong SC, Fang W, Groth P, Zauner KP, Moreau L (2007) Provenance-based validation of e-science experiments. Web Semant Sci Serv Agents World Wide Web 5(1):28–38. https://doi.org/10.1016/j.websem.2006.11.003
29. Wong SC, Miles S, Fang W, Groth P, Moreau L (2005) Provenance-based validation of e-science experiments. In: Gil Y, Motta E, Benjamins VR, Musen MA (eds) The Semantic Web—ISWC 2005. Springer, Heidelberg, pp 801–815. https://doi.org/10.1007/11574620_57
30. Zhao J, Goble C, Greenwood M, Wroe C, Stevens R (2003) Annotating, linking and browsing provenance logs for e-science. In: Ashish N, Goble C (eds) Semantic Web technologies for searching and retrieving scientific data. RWTH Aachen University, Aachen
31. Zhao J, Goble C, Stevens R, Bechhofer S (2004) Semantically linking and browsing provenance logs for e-science. In: Bouzeghoub M, Goble C, Kashyap V, Spaccapietra S (eds) Semantics of a networked world. Springer, Heidelberg, pp 158–176. https://doi.org/10.1007/978-3-540-30145-5_10
32. Zhao J, Wroe C, Goble C, Stevens R, Quan D, Greenwood M (2004) Using Semantic Web technologies for representing e-science provenance. In: McIlraith SA, Plexousakis D, van Harmelen F (eds) The Semantic Web—ISWC 2004. Springer, Heidelberg, pp 92–106. https://doi.org/10.1007/978-3-540-30475-3_8
33. Zednik S, Fox P, McGuinness DL (2010) System transparency, or how I learned to worry about meaning and love provenance! In: McGuinness DL, Michaelis JR, Moreau L (eds) Provenance and annotation of data and processes. Springer, Heidelberg, pp 165–173. https://doi.org/10.1007/978-3-642-17819-1_19
34. Zhao J, Sahoo SS, Missier P, Sheth A, Goble C (2011) Extending semantic provenance into the Web of Data. IEEE Int Comput 15(1):40–48. https://doi.org/10.1109/MIC.2011.7
35. Frey J, Roure DD, Taylor K, Essex J, Mills H, Zaluska E (2006) CombeChem: a case study in provenance and annotation using the Semantic Web. In: Moreau L, Foster I (eds) Provenance and annotation of data. Springer, Heidelberg, pp 270–277. https://doi.org/10.1007/11890850_27
36. Halaschek-Wiener C, Golbeck J, Schain A, Grove M, Parsia B, Hendler J (2006) Annotation and provenance tracking in Semantic Web photo libraries. In: Moreau L, Foster I (eds) Provenance and annotation of data. Springer, Heidelberg, pp 82–89. https://doi.org/10.1007/11890850_10
37. Bunnell L, Osei-Bryson KM, Yoon VY (2019) RecSys issues ontology: a knowledge classification of issues for recommender systems researchers. Inform Syst Front. https://doi.org/10.1007/s10796-019-09935-9
38. Dividino R, Gröner G, Scheglmann S, Thimm M (2012) Ranking RDF with provenance via preference aggregation. In: ten Teije A, Völker J, Handschuh S, Stuckenschmidt H, d'Acquin M, Nikolov A, Aussenac-Gilles N, Hernandez N (eds) Knowledge engineering and knowledge management. Springer, Heidelberg, pp 154–163. https://doi.org/10.1007/978-3-642-33876-2_15
39. Philp D, Chan N, Sikos LF (2019) Decision support for network path estimation via automated reasoning. In: Czarnowski I, Howlett RJ, Jain LC (eds) Intelligent decision technologies 2019. Springer, Singapore, pp 335–344. https://doi.org/10.1007/978-981-13-8311-3_29
40. Sikos LF, Stumptner M, Mayer W, Howard C, Voigt S, Philp D (2018) Automated reasoning over provenance-aware communication network knowledge in support of cyber-situational awareness. In: Liu W, Giunchiglia F, Yang B (eds) Knowledge science, engineering and management. Springer, Cham, pp 132–143. https://doi.org/10.1007/978-3-319-99247-1_12
41. Noy N, Rector A, Hayes P, Welty C (2006) Defining n-ary relations on the Semantic Web. https://www.w3.org/TR/swbp-n-aryRelations/. Accessed 29 March 2020
42. Ding L, Finin T, Peng Y, Da Silva P, McGuinness D (2005) Tracking RDF graph provenance using RDF molecules. In: Fourth International Semantic Web Conference, Galway, Ireland, 6–10 November 2015
43. Berners-Lee T (2005) Notation 3 Logic. https://www.w3.org/DesignIssues/N3Logic. Accessed 29 March 2020
44. Berners-Lee T, Connolly D, Kagal L, Scharf Y, Hendler J (2008) N3Logic: a logical framework for the World Wide Web. Theor Pract Log Prog 8(3):249–269. https://doi.org/10.1017/S1471068407003213
45. Dividino R, Sizov S, Staab S, Schueler B (2009) Querying for provenance, trust, uncertainty and other meta knowledge in RDF. Web Semant Sci Serv Agents World Wide Web 7(3):204–219. https://doi.org/10.1016/j.websem.2009.07.004
46. Schueler B, Sizov S, Staab S (2008) Querying for meta knowledge. In: Proceedings of the 17th International Conference on World Wide Web. ACM, New York, pp 625–634. https://doi.org/10.1145/1367497.1367582
47. Udrea O, Udrea O, Subrahmanian VS (2010) Annotated RDF. ACM Trans Comput Logic 11(2):1–41. https://doi.org/10.1145/1656242.1656245
48. Hoffart J, Suchanek FM, Berberich K, Weikum G (2012) YAGO2: a spatially and temporally enhanced knowledge base

from Wikipedia. Artif Intell 194:28–61. https://doi.org/10.1016/j.artint.2012.06.001
49. Hartig O, Thompson B (2014) Foundations of an alternative approach to reification in RDF. arXiv:1406.3399
50. Keskisärkkä R, Blomqvist E, Lind L, Hartig O (2019) RSP-QL*: enabling statement-level annotations in RDF streams. In: E (ed) Semantic systems. The power of AI and knowledge graphs. Springer, Cham, pp 140–155. https://doi.org/10.1007/978-3-030-33220-4_11
51. Damiani E, Oliboni B, Quintarelli E, Tanca L (2019) A graph-based meta-model for heterogeneous data management. Knowl Inf Syst 61(1):107–136. https://doi.org/10.1007/s10115-018-1305-8
52. Straccia U, Lopes N, Lukacsy G, Polleres A (2010) A general framework for representing and reasoning with annotated Semantic Web data. In: Proceedings of the 24th AAAI Conference on Artificial Intelligence. AAAI Press, Menlo Park, pp 1437–1442
53. Zimmermann A, Lopes N, Polleres A, Straccia U (2012) A general framework for representing, reasoning and querying with annotated Semantic Web data. Web Semant Sci Serv Agents World Wide Web 11:72–95. https://doi.org/10.1016/j.websem.2011.08.006
54. Livingston KM, Bada M, Hunter LE, Verspoor K (2013) Representing annotation compositionality and provenance for the Semantic Web. J Biomed Semant 4:38. https://doi.org/10.1186/2041-1480-4-38
55. Sahoo SS, Bodenreider O, Hitzler P, Sheth A, Thirunarayan K (2010) Provenance Context Entity (PaCE): scalable provenance tracking for scientific RDF data. In: Gertz M, Ludäscher B (eds) Scientific and statistical database management. Springer, Heidelberg, pp 461–470. https://doi.org/10.1007/978-3-642-13818-8_32
56. Nguyen V, Bodenreider O, Sheth A (2014) Don't like RDF reification? Making statements about statements using singleton property. In: Proceedings of the 23rd International Conference on World Wide Web. ACM, New York, pp 759–770. https://doi.org/10.1145/2566486.2567973
57. Carroll JJ, Bizer C, Bizer C, Stickler P (2005) Named graphs, provenance and trust. In: Proceedings of the 14th International Conference on World Wide Web, ACM, New York, pp 613–622. https://doi.org/10.1145/1060745.1060835
58. Watkins ER, Nicole DA (2006) Named graphs as a mechanism for reasoning about provenance. In: Zhou X, Li J, Shen HT, Kitsuregawa M, Zhang Y (eds) Frontiers of WWW Research and Development—APWeb 2006. Springer, Heidelberg, pp 943–948. https://doi.org/10.1007/11610113_99
59. Pediaditis P, Flouris G, Fundulaki I, Christophides V (2009) On explicit provenance management in RDF/S graphs. First Workshop on the Theory and Practice of Provenance, San Francisco, CA, USA, 23 February 2009
60. Flouris G, Fundulaki I, Pediaditis P, Theoharis Y, Christophides V (2009) Coloring RDF triples to capture provenance. In: Bernstein A, Karger DR, Heath T, Feigenbaum L, Maynard D, Motta E, Thirunarayan K (eds) The Semantic Web—ISWC 2009. Springer, Heidelberg, pp 196–212. https://doi.org/10.1007/978-3-642-04930-9_13
61. Groth P, Gibson A, Velterop J (2010) The anatomy of a nanopublication. Inform Serv Use 30(1–2):51–56. https://doi.org/10.3233/ISU-2010-0613
62. Hogan A (2018) Context in graphs. In: Proceedings of the 1st International Workshop on Conceptualized Knowledge Graphs. RWTH Aachen University, Aachen
63. Sikos LF, Stumptner M, Mayer W, Howard C, Voigt S, Philp D (2018) Representing network knowledge using provenance-aware formalisms for cyber-situational awareness. Procedia Comput Sci 126:29–38. https://doi.org/10.1016/j.procs.2018.07.206
64. Analyti A, Damásio CV, Antoniou G, Pachoulakis I (2014) Why-provenance information for RDF, rules, and negation. Ann Math Artif Intell 70(3):221–277. https://doi.org/10.1007/s10472-013-9396-0
65. Gutierrez C, Hurtado CA, Vaisman A (2007) Introducing time into RDF. IEEE T Knowl Data Eng 19(2):207–218. https://doi.org/10.1109/TKDE.2007.34
66. Hurtado C, Vaisman A (2006) Reasoning with temporal constraints in RDF. In: Alferes JJ, Bailey J, May W, Schwertel U (eds) Principles and practice of Semantic Web reasoning. Springer, Heidelberg, pp 164–178. https://doi.org/10.1007/11853107_12
67. Tappolet J, Bernstein A (2009) Applied temporal RDF: efficient temporal querying of RDF data with SPARQL. In: Aroyo L, Traverso P, Ciravegna F, Cimiano P, Heath T, Hyvönen E, Mizoguchi R, Oren E, Sabou M, Simperl E (eds) The Semantic Web: research and applications. Springer, Heidelberg, pp 308–322. https://doi.org/10.1007/978-3-642-02121-3_25
68. McGrath R, Futrelle J (2008) Reasoning about provenance with OWL and SWRL rules. In: AAAI 2008 Spring Symposia, Palo Alto, CA, USA, 26–28 March 2008
69. Aljalbout S, Buchs D, Falquet G (2019) Introducing contextual reasoning to the Semantic Web with OWL^C. In: Endres D, Alam M, Şotropa D (eds) Graph-based representation and reasoning. Springer, Cham, pp 13–26. https://doi.org/10.1007/978-3-030-23182-8_2
70. Zimmermann A, Giménez-García JM (2017) Integrating context of statements within description logics. arXiv:1709.04970v1
71. Klarman S (2013) Reasoning with contexts in description logics. Ph.D. thesis, VU University Amsterdam, Amsterdam, Netherlands
72. Tarski A (1944) The semantic conception of truth and the foundations of semantics. Philos Phenomen Res 4(3):341–376
73. Hayes P, Patel-Schneider P (2014a) RDF 1.1 semantics. https://www.w3.org/TR/rdf11-mt/. Accessed 29 March 2020
74. Sikos LF (2017) Description logics in multimedia reasoning. Springer, Cham. https://doi.org/10.1007/978-3-319-54066-5
75. Hayes P, Patel-Schneider P (2014) Simple Interpretations. In: RDF 1.1 semantics. https://www.w3.org/TR/rdf11-mt/#simple-interpretations. Accessed 29 March 2020
76. Hayes P, Patel-Schneider P (2014) RDFS interpretations. In: RDF 1.1 semantics. https://www.w3.org/TR/rdf11-mt/#rdfs-interpretations. Accessed 29 March 2020
77. Gardenfors P (1992) The dynamics of belief systems: foundations versus coherence theories. Rev Int Philos 44:24–46
78. Newman A, Li Y, Hunter J (2008) A scale-out RDF molecule store for improved coidentification, querying and inferencing. In: International Workshop on Scalable Semantic Web Knowledge Base Systems, Beijing, China, 22 April 2008
79. Zhao J, Miles A, Klyne G, Shotton D (2008) Linked Data and provenance in biological data Webs. Brief Bioinform 10(2):139–152. https://doi.org/10.1093/bib/bbn044
80. da Silva PP, McGuinness DL, Fikes R (2006) A proof markup language for Semantic Web services. Inform Syst 31(4–5):381–395. https://doi.org/10.1016/j.is.2005.02.003
81. Ding L, Bao J, Michaelis JR, Zhao J, McGuinness DL (2010) Reflections on provenance ontology encodings. In: McGuinness DL, Michaelis JR, Moreau L (eds) Provenance and annotation of data and processes. Springer, Heidelberg, pp 198–205. https://doi.org/10.1007/978-3-642-17819-1_22
82. Lebo T, P W, Graves A, McGuinness D (2012) Towards unified provenance granularities. In: Groth P, Frew J (eds) Provenance and annotation of data and processes. Springer, Heidelberg, pp 39–51. https://doi.org/10.1007/978-3-642-34222-6_4
83. Moreau L, Groth P, Cheney J, Lebo T, Miles S (2015) The rationale of PROV. Web Semant Sci Serv Agents World Wide Web 35(4):235–257. https://doi.org/10.1016/j.websem.2015.04.001

84. McGuinness D, Ding L, da Silva P, Chang C (2007) PML 2: a modular explanation interlingua. In: Roth-Berghofer T, Schulz S, Bahls D, Leake D (eds) Explanation-aware computing. AAAI Press, Menlo Park, pp 49–55
85. Sahoo S, Sheth A (2009) Provenir ontology: towards a framework for eScience provenance management. Microsoft eScience Workshop, Pittsburgh, PA, USA, 15–17 October 2009
86. Missier P, Sahoo SS, Zhao J, Goble C, Sheth A (2010) Janus: from workflows to semantic provenance and Linked Open Data. In: McGuinness DL, Michaelis JR, Moreau L (eds) Provenance and annotation of data and processes. Springer, Heidelberg, pp 129–141. https://doi.org/10.1007/978-3-642-17819-1_16
87. Ciccarese P, Soiland-Reyes S, Belhajjame K, Gray AJG, Goble C, Clark T (2013) PAV ontology: provenance, authoring and versioning. J Biomed Semant 4:37. https://doi.org/10.1186/2041-1480-4-37
88. Da Cruz S, Campos M, Mattoso M (2012) A foundational ontology to support scientific experiments. In: Malucelli A, Bax M (eds) Proceedings of Joint V Seminar on Ontology Research in Brazil and VII International Workshop on Metamodels, Ontologies and Semantic Technologies. RWTH Aachen University, Aachen, pp 144–155
89. Di Iorio A, Caron B (2016) PREMIS 3.0 ontology: Improving semantic interoperability of preservation metadata. In: Proceedings of the 13th International Conference on Digital Preservation. Swiss National Library, Bern, pp 32–36
90. Hartig O, Zhao J (2010) Publishing and consuming provenance metadata on the Web of Linked Data. In: McGuinness DL, Michaelis JR, Moreau L (eds) Provenance and annotation of data and processes. Springer, Heidelberg, pp 78–90. https://doi.org/10.1007/978-3-642-17819-1_10
91. Moreau L, Clifford B, Freire J, Futrelle J, Gil Y, Groth P, Kwasnikowska N, Miles S, Missier P, Myers J, Plale B, Simmhan Y, Stephan E, den Bussche JV (2011) The Open Provenance Model Core Specification (v1.1). Future Gener Comp Sy 27(6):743–756. https://doi.org/10.1016/j.future.2010.07.005
92. Anam S, Kang B, Kim Y, Liu Q (2015) Linked Data provenance: state of the art and challenges. In: 3rd Australasian Web Conference, Sydney, Australia, 27–30 January 2015
93. Omitola T, Omitola T, Gutteridge C, Millard IC, Glaser H, Gibbins N, Shadbolt N (2011) Tracing the provenance of Linked Data using VoID. In: Akerkar R (ed) Proceedings of the International Conference on Web Intelligence, Mining and Semantics. https://doi.org/10.1145/1988688.1988709
94. Lagoze C, Van de Sompel H, Johnston P, Nelson M, Sanderson R, Warner S (2008) ORE user guide–resource map implementation in RDF/XML. http://www.openarchives.org/ore/1.0/rdfxml. Accessed 29 March 2020
95. Garijo D, Eckert K, Miles S, Trim C, Panzer M (2013) Dublin Core to PROV mapping. https://www.w3.org/TR/prov-dc/. Accessed 29 March 2020
96. Welty C, Fikes R (2006) A reusable ontology for fluents in OWL. In: Proceedings of the 2006 Conference on Formal Ontology in Information Systems. IOS Press, Amsterdam, pp 226–236
97. Giménez-García JM, Zimmermann A, Maret P (2017) NdFluents: an ontology for annotated statements with inference preservation. In: Blomqvist E, Maynard D, Gangemi A, Hoekstra R, Hitzler P, Hartig O (eds) The Semantic Web. Springer, Cham, pp 638–654. https://doi.org/10.1007/978-3-319-58068-5_39
98. Liu L, Özsu M (2018) Encyclopedia of database systems, 2nd edn. Springer, New York
99. Vicknair C, Macias M, Zhao Z, Nan X, Chen Y, Wilkins D (2010) A comparison of a graph database and a relational database: a data provenance perspective. In: Proceedings of the 48th Annual Southeast Regional Conference. ACM, New York. https://doi.org/10.1145/1900008.1900067
100. Chebotko A, Abraham J, Brazier P, Piazza A, Kashlev A, Lu S (2013) Storing, indexing and querying large provenance data sets as RDF graphs in Apache HBase. In: IEEE Ninth World Congress on Services, IEEE, New York. https://doi.org/10.1109/SERVICES.2013.32
101. Wylot M, Cudre-Mauroux P, Groth P (2014) TripleProv: Efficient processing of lineage queries in a native RDF store. In: Proceedings of the 23rd International Conference on World Wide Web. ACM, New York, pp 455–466. https://doi.org/10.1145/2566486.2568014
102. Wylot M, Cudré-Mauroux P, Groth P (2015) Adaptive RDF query processing based on provenance. In: Ludäscher B, Plale B (eds) Provenance and annotation of data and processes. Springer, Cham, pp 264–266. https://doi.org/10.1007/978-3-319-16462.-5_29
103. Brauer PC, Fittkau F, Hasselbring W (2015) The aspect-oriented architecture of the CAPS Framework for capturing, analyzing and archiving provenance data. In: Ludäscher B, Plale B (eds) Provenance and annotation of data and processes. Springer, Cham, pp 223–225. https://doi.org/10.1007/978-3-319-16462-5_19
104. OpenLink Software (2017) Do you support additional metadata for triples, such as time-stamps, security tags etc? In: Openlink Virtuoso Universal Server Documentation. http://docs.openlinksw.com/virtuoso/virtuosofaq13/. Accessed 29 March 2020
105. Erling O (2018) Provenance and reification in Virtuoso. https://www.openlinksw.com/weblog/oerling/?id=1572. Accessed 29 March 2020
106. Philp D, Thomas L, Gilmartin D, Voigt S, (2018) Cyber situational awareness for communication networks. AOC, (2018) Convention. Adelaide, Australia
107. Dimou A, De Nies T, Verborgh R, Mannens E, Van de Walle R (2016) Automated metadata generation for Linked Data generation and publishing workflows. In: Auer S, Berners-Lee T, Bizer C, Heath T (eds) Proceedings of the 9th Workshop on Linked Data on the Web. RWTH Aachen University, Aachen
108. De Mendonça R, da Cruz S, De La Cerda J, Cavalcanti M, Cordeiro K, Campos M (2013) LOP: capturing and linking open provenance on LOD cycle. In: Proceedings of the Fifth Workshop on Semantic Web Information Management. ACM, New York. https://doi.org/10.1145/2484712.2484715
109. Beek W, Raad J, Wielemaker J, van Harmelen F (2018) sameAs.cc: the closure of 500M owl:sameAs statements. In: Gangemi A, Navigli R, Vidal ME, Hitzler P, Troncy R, Hollink L, Tordai A, Alam M (eds) The Semantic Web. Springer, Cham, pp 65–80. https://doi.org/10.1007/978-3-319-93417-4_5
110. McCusker J, McGuinness D (2010) owl:sameAs considered harmful to provenance. In: ISCB Conference on Semantics in Healthcare and Life Sciences, Cambridge, MA, USA, 2010
111. Sikos LF, Philp D, Voigt S, Howard C, Stumptner M, Mayer W (2018a) Provenance-aware LOD datasets for detecting network inconsistencies. In: Proceedings of the 1st International Workshop on Conceptualized Knowledge Graphs. RWTH Aachen University, Aachen
112. Dezani-Ciancaglini M, Horne R, Sassone V (2012) Tracing where and who provenance in Linked Data: a calculus. Theor Comput Sci 464:113–129. https://doi.org/10.1016/j.tcs.2012.06.020
113. Eckert K, Garijo D, Panzer M, Percin O (2011) Metadata provenance: Dublin Core on the next level. In: Baker T, Hillmann D, Isaac A (eds) Proceedings of the International Conference on Dublin Core and Metadata Applications, The Hague, The Netherlands, 21–23 September 2011
114. Freitas A, Legendre A, O'Riain S, Curry E (2010) Prov4J: a Semantic Web framework for generic provenance management. In: Sahoo S, Zhao J, Missier P, Gomez-Perez J (eds) Proceedings of the Second International Workshop on the Role of Seman-

tic Web in Provenance Management. RWTH Aachen University, Aachen
115. Trinh TD, Aryan P, Do BL, Ekaputra F, Kiesling E, Rauber A, Wetz P, Tjoa A (2017) Linked Data processing provenance: towards transparent and reusable Linked Data integration. In: Proceedings of the International Conference on Web Intelligence. ACM, New York, pp 88–96. https://doi.org/10.1145/3106426.3106495
116. Sharma K, Marjit U, Biswas U (2015) Efficient provenance storage for RDF dataset in Semantic Web environment. In: 2015 International Conference on Information Technology. IEEE, New York. https://doi.org/10.1109/ICIT.2015.21
117. Wylot M (2015) Efficient, scalable, and provenance-aware management of Linked Data. Ph.D. thesis, University of Fribourg, Fribourg, Switzerland
118. Ding L, Michaelis J, McCusker J, McGuinness D (2011) Linked Provenance Data: a Semantic Web-based approach to interoperable workflow traces. Future Gener Comput Syst 27(6):797–805. https://doi.org/10.1016/j.future.2010.10.011
119. Wylot M, Cudre-Mauroux P, Groth P (2015) A demonstration of TripleProv: tracking and querying provenance over Web data. VLDB Endowment 8(12):1992–1995 https://doi.org/10.14778/2824032.2824119
120. Wylot M, Cudre-Mauroux P, Groth P (2015) Executing provenance-enabled queries over web data. In: Proceedings of the 24th International Conference on World Wide Web. Springer, Heidelberg, pp 1275–1285. https://doi.org/10.1145/2736277.2741143
121. Damásio CV, Analyti A, Antoniou G (2012) Provenance for SPARQL queries. In: Cudré-Mauroux P, Heflin J, Sirin E, Tudorache T, Euzenat J, Hauswirth M, Parreira JX, Hendler J, Schreiber G, Bernstein A, Blomqvist E (eds) The Semantic Web—ISWC 2012. Springer, Heidelberg, pp 625–640. https://doi.org/10.1007/978-3-642-35176-1_39
122. Halpin H, Cheney J (2014) Dynamic provenance for SPARQL updates using named graphs. In: Proceedings of the 23rd International Conference on World Wide Web. ACM, New York, pp 287–288. https://doi.org/10.1145/2567948.2577357
123. Halpin H, Cheney J (2014) Dynamic provenance for SPARQL updates. In: Mika P, Tudorache T, Bernstein A, Welty C, Knoblock C, Vrandečić D, Groth P, Noy N, Janowicz K, Goble C (eds) The Semantic Web—ISWC 2014. Springer, Cham, pp 425–440. https://doi.org/10.1007/978-3-319-11964-9_27
124. Avgoustaki A, Flouris G, Fundulaki I, Plexousakis D (2016) Provenance management for evolving RDF datasets. In: Sack H, Blomqvist E, d'Aquin M, Ghidini C, Ponzetto SP, Lange C (eds) The Semantic Web. Latest advances and new domains. Springer, Cham, pp 575–592. https://doi.org/10.1007/978-3-319-34129-3_35
125. Geerts F, Unger T, Karvounarakis G, Fudulaki I, Christophides V (2016) Algebraic structures for capturing the provenance of SPARQL queries. J ACM 63:1–63. https://doi.org/10.1145/2810037
126. Sikos LF (2016) RDF-powered semantic video annotation tools with concept mapping to Linked Data for next-generation video indexing. Multim Tools Appl 76(12):14437–14460. https://doi.org/10.1007/s11042-016-3705-7
127. Patton EW, Difranzo D, McGuinness DL (2010) SAF: a provenance-tracking framework for interoperable semantic applications. In: McGuinness DL, Michaelis JR, Moreau L (eds) Provenance and annotation of data and processes. Springer, Heidelberg, pp 73–77. https://doi.org/10.1007/978-3-642-17819-1_9
128. Narock T, Yoon V, March S (2014) A provenance-based approach to Semantic Web service description and discovery. J Decis Support Syst 64(C):90–99. https://doi.org/10.1016/j.dss.2014.04.007
129. Michaelis J, McGuinness D (2010) Towards provenance aware comment tracking for Web applications. In: McGuinness DL, Michaelis JR, Moreau L (eds) Provenance and annotation of data and processes. Springer, Heidelberg, pp 265–273. https://doi.org/10.1007/978-3-642-17819-1_30
130. Hunter J, Cheung K (2007) Provenance Explorer–a graphical interface for constructing scientific publication packages from provenance trails. Int J Digit Libr 7(1–2):99–107. https://doi.org/10.1007/s00799-007-0018-5
131. Alper P, Belhajjame K, Goble CA, Karagoz P (2015) LabelFlow: exploiting workflow provenance to surface scientific data provenance. In: Ludäscher B, Plale B (eds) Provenance and annotation of data and processes. Springer, Cham, pp 84–96. https://doi.org/10.1007/978-3-319-16462-5_7
132. Macko P, Seltzer M (2012) A general-purpose provenance library. In: Proceedings of the 4th USENIX Conference on Theory and Practice of Provenance
133. Myers J, Futrelle J, Gaynor J, Plutchak J, Bajcsy P, Kastner J, Kotwani K, Lee J, Marini L, Kooper R, McGrath R, McLaren T, Rodriguez A, Liu Y (2008) Embedding data within knowledge spaces. arXiv:0902.0744
134. Gerber D, Esteves D, Lehmann J, Bühmann L, Usbeck R, Ngomo ACN, Speck R (2015) DeFacto–temporal and multilingual deep fact validation. Web Semant Sci Serv Agents World Wide Web 35:85–101. https://doi.org/10.1016/j.websem.2015.08.001
135. Wang X, Wang J (2016) ProvRPQ: an interactive tool for provenance-aware regular path queries on RDF graphs. In: Cheema MA, Zhang W, Chang L (eds) Databases theory and applications. Springer, Cham, pp 480–484. https://doi.org/10.1007/978-3-319-46922-5_44
136. Philp D, Chan N, Mayer W (2019) Network path estimation in uncertain data via entity resolution. In: Le TD, Ong KL, Zhao Y, Jin WH, Wong S, Liu L, Williams G (eds) Data mining. Springer, Singapore, pp 196–207. https://doi.org/10.1007/978-981-15-1699-3_16
137. Turnbull B, Randhawa S (2015) Automated event and social network extraction from digital evidence sources with ontological mapping. Digit Invest 13:94–106. https://doi.org/10.1016/j.diin.2015.04.004
138. Lopes N, Kirrane S, Zimmermann A, Polleres A, Mileo A (2012) A logic programming approach for access control over RDF. In: Dovier A, Costa VS (eds) Technical communications of the 28th International Conference on Logic Programming (ICLP'12). Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, Dagstuhl, pp 381–392. https://doi.org/10.4230/LIPIcs.ICLP.2012.381
139. Sorici A, Picard G, Boissier O, Zimmermann A, Florea A (2015) CONSERT: applying Semantic Web technologies to context modeling in ambient intelligence. Comput Electr Eng 44:280–306. https://doi.org/10.1016/j.compeleceng.2015.03.012
140. Sharma B, Keshan N, Agu N, Chari S, Narkar S (2019) Diabetes treatment support ontology. http://tw.rpi.edu/media/latest/DiabetesTreatmentSupport_DraftProjectPaper.pdf. Accessed 29 March 2020
141. Nguyen V, Sheth AP (2017) Logical inferences with contexts of RDF triples. arXiv:1701.05724
142. Klinov P (2017) How to read Stardog query plans. https://www.stardog.com/blog/how-to-read-stardog-query-plans/. Accessed 29 March 2020