# Evolution of Workflow Provenance Information in the Presence of Custom Inference Rules

Christos Strubulis, Yannis Tzitzikas, Martin Doerr and Giorgos Flouris

Institute of Computer Science, FORTH-ICS, GREECE, and
Computer Science Department, University of Crete, GREECE
{strubul|tzitzik|martin|fgeo}@ics.forth.gr

**Abstract.** Workflow systems can produce very large amounts of provenance information. In this paper we introduce provenance-based inference rules as a means to reduce the amount of provenance information that has to be stored, and to ease quality control (e.g., corrections). We motivate this kind of (provenance) inference and identify a number of basic inference rules over a conceptual model appropriate for representing provenance. The proposed inference rules concern the interplay between (i) actors and carried out activities, (ii) activities and devices that were used for such activities, and, (iii) the presence of information objects and physical things at events. However, since a knowledge base is not static but it changes over time for various reasons, we also study how we can satisfy change requests while supporting and respecting the aforementioned inference rules. Towards this end, we elaborate on the specification of the required change operations.

## 1 Introduction

Workflow systems can produce huge amounts of provenance information. For example, in empirical 3D model generation, where tens of thousands of intermediate files and processes of hundreds of individual manual actions are no rarity, it is prohibitive to register each item's complete history because of the immense repetition of facts: on the one side, the storage space needed would be blown up by several orders of magnitude, and, on the other, any correction of erroneous input would require tracing the huge proliferation graph of this input. In this paper we introduce *provenance-based inference rules* as a means to reduce the amount of provenance information that has to be stored, and to ease quality control (e.g., corrections). Note that the above notion of redundancy is yet formally not well understood and may even not be strictly logical. For instance, it is a question of convention whether we regard that persons carrying out an activity carry out all of its subactivities.

In this paper we consider CRMdig [24] as the conceptual model for representing provenance, and over this model we identify custom inference rules. The identified inference rules concern the interplay between (i) actors and carried out activities, (ii) activities and devices that were used in such activities, and, (iii) the presence of information objects and physical things at events. We focus on these particular rules because the classes involved belong to almost every provenance model and they occur frequently in practice. Of course, one could extend this set according to the details and conventions of the application at hand. In general, we could say that the major sources of inference are: transitivity of part-hood relations and propagation of properties from wholes to parts, be it for objects and their parts or for processes and their subprocesses.

Supporting these rules raises complications when a knowledge base (either stored in a system or composed by various metadata files) changes over time, as we need to satisfy change requests while still supporting the aforementioned inference rules. To tackle the update requirement we elaborate on the specification of change operations which handle changes while respecting the aforementioned inference rules. Specifically, we propose three operations, namely *Add, Disassociate,* and *Contract,* and discuss their semantics and application in our setting.

The rest of this paper is organized as follows: Section 2 discusses in brief our application context and assumptions; Section 3 introduces the provenance inference rules; subsequently, Section 4 elaborates on the knowledge evolution requirements and their interplay with the inference rules; Section 5 discusses related work, and finally Section 6 concludes the paper and identifies issues that are worth further research.

## 2 Background, Context and Working Assumptions

### 2.1 Application Context

There are several models for representing provenance. In this work we consider CRMdig [24] a structurally object-oriented model which is an extension of the CIDOC CRM ontology (ISO 21127:2006) [7]. In brief, CIDOC CRM is a core ontology describing the underlying semantics of data schemata and structures from all museum disciplines and archives. It is the result of a long-term interdisciplinary work and agreement and it has been derived by integrating (in a bottom-up manner) hundreds of metadata schemas. CRMdig was initially defined during the EU Project CASPAR[1] (FP6-2005-IST-033572) and its evolution continues in the context of the EU IST IP 3D-COFORM[2] project. In numbers, CIDOC CRM contains 86 classes and 137 properties, while its extension CRMdig currently contains 31 classes and 70 properties. Fig. 1 shows one small part of the model, specifically the part related to the inference rules which are introduced in Section 3.

The shown properties and classes are described in detail in CIDOC CRM's official definition.[3] In brief, the properties "is composed of" and "forms part of" represent the part-hood relationships of man-made objects (i.e., instances of the "Man-made Object" class) and activities (i.e., instances of the "Activity" class) respectively. Fur-

[1] http://www.casparpreserves.eu/
[2] http://www.3d-coform.eu/
[3] http://www.cidoc-crm.org/docs/cidoc_crm_version_5.0.4.pdf

thermore, the property “carried out by” describes the active participation of actors (i.e., instances of the “Actor” class) in activities and also implies causal or legal responsibility. Moreover, “was used for” describes the use of objects in a way essential to the performance of an activity. Finally, immaterial items (i.e., instances of the “Information Object” class) are related to physical carriers (i.e., instances of the “Physical Man-Made Thing” class) via the “carries” property and can be present at events (i.e., instances of the “Event” class) via the “was present at” property. Note that the properties “is composed of” and “forms part of” are transitive, reflexive and anti-symmetric.

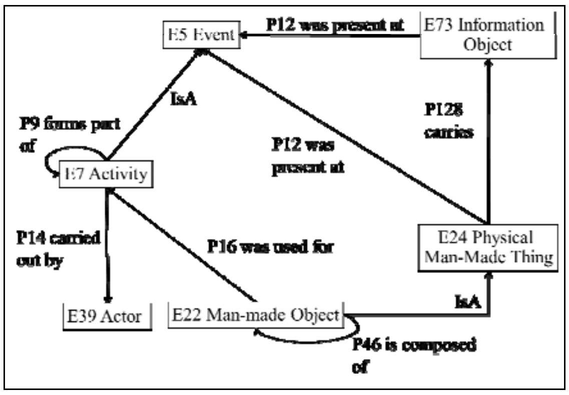

Fig. 1. Part of CRM Dig

## 2.2 Working Assumptions

We shall use the term *KB* to refer to a Knowledge Base in the logical sense, either stored in a system or composed by the contents of several metadata files, and *K* to refer to the set of RDF/S triples of a KB. Furthermore, hereafter we assume that *K* denotes the set of triples produced by the adopted metadata schemas and ontologies and the ingested facts (yielded by manual or automated processes).

We shall use *C(K)* to refer to the *closure* of *K*. Note that the closure can be defined with respect to the standard inference rules for RDF/S and/or other custom rules (e.g., like those in Section 3) and is unique. We shall also use *Red(K)* to denote the *reduction* of *K*, which is the minimal set of facts that have the same closure as *K*, and it is unique if and only if the relations are acyclic [25].

A repository policy could be to keep stored either *K*, or *Red(K)*, or *C(K)*. Storing *Red(K)* would give optimum (i.e., minimal) space usage, but would imply an important overhead during changes, because the reduction should be recalculated after every change. Though, *C(K)* is optimal with respect to efficiency, because all the information is stored and can be easily found, but has increased space requirements. For this reason, we chose to store *K* as a reasonable compromise in this time-space tradeoff and compute *C(K)* dynamically e.g., at query time. Note that, in practice, *K* usually contains little or no redundancy, leading to a near-optimal space usage while avoiding the overhead of searching for, and eliminating redundancy.

## 3 Provenance Inference Rules

The proposed inference rules concern mainly the three binary relations, namely "carriedOutBy(Actor, Activity)", "wasUsedFor(Device, Activity)", and "wasPresentAt(InformationObject, Event)". The corresponding rules are intuitively defined as:

- R1: If an actor has carried out one activity, then he has carried out all of its subactivities.
- R2: If an object (device) was used for an activity, then all parts of the object were used for that activity too.
- R3: If a physical thing that carries an information object was present at an event, then that information object was present at that event too.

More formally, the above three rules can be encoded into first-order logic (FOL) using the above relations (see also Fig. 1) as follows:

- R1: $\forall$x,y,z ( formsPartOf (y,x) $\wedge$ carriedOutBy (x,z)$\rightarrow$ carriedOutBy (y,z) )
- R2: $\forall$x,y,z ( isComposedOf (x,y) $\wedge$ wasUsedFor (x,z)$\rightarrow$ wasUsedFor (y,z) )
- R3: $\forall$x,y,z ( carries (x,y) $\wedge$ wasPresentAt (x,z)$\rightarrow$ wasPresentAt(y,z) )

Subsequently we provide examples for each rule. In the included figures we do not show the transitivity-induced properties for "is composed of" and "forms part of":
**Rule 1:** scientists often use 3D laser scanning to construct digital 3D models by taking many photographs of the desired model. One example from our data is the activity *Laser scanning acquisition of Canoe-shaped vase from Archaeological Museum of Nicosia* which was carried out by the *STARC-The Cyprus Institute*. The activity can be analyzed to *Sequence of shots* and to the subactivities: *Capture 1_7* and *Capture 1_8* storing different recorded metadata about each photo. However, the information that the initial actor was the *Starc Institute* is desired to be preserved following the path. Fig. 2 shows the edges (represented by dotted lines) inferred by the rule.

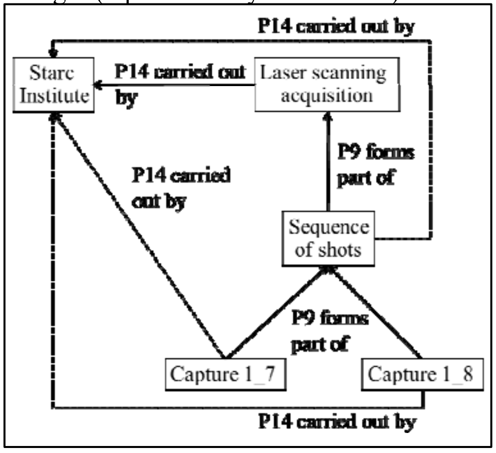

Fig. 2. Example of rule R1

**Rule 2:** in 3D modeling, devices with many cameras, called multiviewdome devices, are usually employed. These devices have as parts other cameras or lighting devices. Consider a fact stating that a multiviewdome device was used for that activity. With R2 we can infer that the constituent devices were also used for that activity. Indeed, a multiviewdome device cannot be used without using its parts. Fig. 3 illustrates an indicative modeling of such a setting.

**Rule 3:** consider the exhibit shown in Fig. 4 (left) which is part of a column of Ramesses II located in the Egyptian museum garden in Cairo. The 3D reconstruction process of said exhibit could be modeled as an *event* and the exhibit itself as a *physical man-made thing*. Moreover, the carved hieroglyphics could be modeled as an *information object*. Rule R3 infers the presence of information in hieroglyphics at the event of a 3D reconstruction because the part of Ramesses II was also present at that event (see Fig. 4 (right)). The inference is reasonable because the information was carved in hieroglyphics when the column was built, and this implies not only its coexistence with the column but also its presence at the same events.

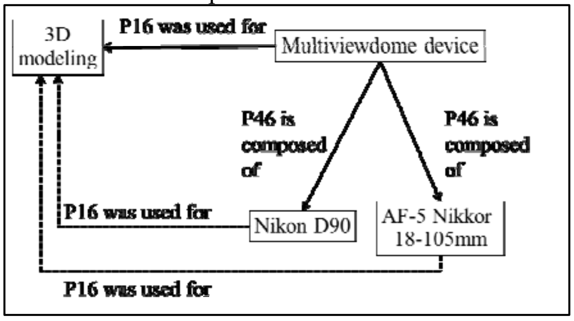

Fig. 3. Example of rule R2

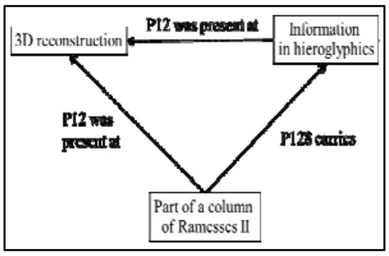

Fig. 4. Part of a column of Ramesses II (left) example of Rule R3 (right)

## 4 Provenance Inference Rules and Knowledge Evolution

A KB changes over time, i.e., we may have requests for adding or deleting facts. Satisfying update requests while still supporting the inference rules is a challenging issue, because several problems arise when updating knowledge taking into account rules and implicit facts. These issues are described below with a running example. For each change operation, we describe the KB's states (through figures) and explain the challenges incurred in the update process due to the existence of the inference rules.

Consider a KB that contains the example with the activities of *Laser scanning acquisition* that were carried out by the *Starc Institute.* The initial state of the KB is demonstrated in Fig. 5 (left). Over this example we shall apply three change operations: *addition, disassociation,* and *contraction.*

**Adding an actor carrying out an activity** (addition of a "carried out by" relationship): suppose a request that *Michael* (a photographer) is also the actor of *Sequence of Shots*. The update is demonstrated in the following figures. We observe that *Michael* has been associated with the activity *Sequence of shots*, but also, due to rule R1, he has been (implicitly) associated with the subactivities *Capture 1_7* and *Capture 1_8*. Note that the propagation of actors from subactivities to superactivities is not assumed and defined as a rule. An actor's responsibility for a subactivity does not imply his responsibility for its superactivity and as a result for the rest subactivities of the latter.

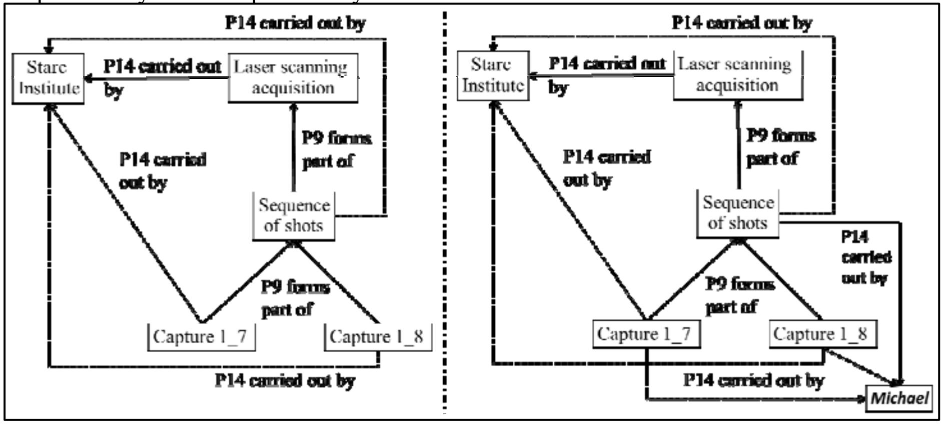

Fig. 5. Initial state of the KB (left) and state of the KB after the addition (right)

**Deleting an actor from carrying out an activity** (disassociation or contraction of a "carried out by" relationship): the problem becomes more complicated when dealing with the deletion of facts. Consider an update request saying that the *Starc Institute* is not responsible for the activity *Capture 1_7*. The rising question is whether the *Starc Institute* is not responsible only for *Capture 1_7*, or also for other activities, i.e., how refuting the fact that it is responsible for *Capture 1_7* affects the information that it is responsible for *Laser scanning acquisition*, *Sequence of shots*, or *Capture 1_8*.

Initially, we note that the *Starc Institute* should also be disassociated from the responsibility of *Sequence of shots* and *Laser scanning acquisition*; failing to do so would cause the subsequent undesirable re-emergence of the refuted knowledge (i.e., that the *Starc Institute* is not responsible for *Capture 1_7*) due to inference.

A more complicated issue is whether the *Starc Institute* should remain responsible for *Capture 1_8*: note that this information was originally included just because of the fact that the *Starc Institute* was considered responsible for *Laser scanning acquisition*, ergo (due to the inference rules), also responsible for *Capture 1_8*. Once the former is dropped, as discussed above, it is questionable whether the latter (inferred) information should still remain in the KB, since its "reason for existence" is no longer there. On the other hand, the fact that the *Starc Institute* is not responsible for *Capture 1_7* does not in any way exclude the possibility that it is still responsible for *Capture 1_8*, therefore deleting this information seems like an unnecessary loss of knowledge.

To address this issue, one should go deeper and study the related philosophical issues regarding the epistemological status of the inferred knowledge, and whether such knowledge has the same or different value compared to primary, explicitly provided knowledge (i.e., ingested knowledge). There are two viewpoints in this respect: *foundational theories* and *coherence theories* [10].

Under the *foundational* viewpoint, each piece of our knowledge serves as a justification for other beliefs. This viewpoint implies that the ingested facts are more important than other knowledge and that implicit knowledge has no value of its own, but is depending on the support of the explicit knowledge that caused its inference.

On the other hand, under the *coherence* theory, no justification is required for our knowledge; each piece of knowledge is justified by how well it fits with the rest of the knowledge, in forming a coherent set of facts that contains no contradictions. This means that all knowledge (implicit or explicit) has the same "value" and that every piece of knowledge (including implicit ones) is self-justified and needs no support from explicit knowledge.

This distinction is vital for effective management of data deletions. When a piece of knowledge is deleted, all implicit data that is no longer supported must be deleted as well under the foundational viewpoint. In our example, this should cause the deletion of the fact that the *Starc Institute* is responsible for *Capture 1_8*. On the other hand, the coherence viewpoint will only delete implicit data if it contradicts with existing knowledge, because the notion of support is not relevant for the coherence model. Therefore, in our case, the fact that the *Starc Institute* is responsible for *Capture 1_8* should persist, because it does not in any way contradict the rest of our knowledge, nor does it cause the re-emergence of the recently deleted information.

Instead of positioning ourselves in favour of one or the other approach, we decided to support both. This is done by defining two different "deletion" operations, namely, *disassociation* and *contraction*, that allow us to support both viewpoints:

**Disassociation:** disassociation handles deletion using the foundational viewpoint. In our example, the non-responsibility of the *Starc Institute* about *Capture 1_7* implies some uncertainty about its responsibility for other related activities (i.e., was he responsible for *Capture 1_8*?), since this knowledge is no longer supported by any explicit data. Based on the foundational viewpoint, all such associations must also be deleted, i.e., we should delete the following:

- (Capture 1_7, carried out by, Starc Institute) // as requested
- (Capture 1_8, carried out by, Starc Institute)
- (Sequence of shots, carried out by, Starc Institute)
- (Laser scanning acquisition, carried out by, Starc Institute)

In practice, implicit facts are not stored so need not be deleted; thus, in our case, we only need to delete: (Laser scanning acquisition, carried out by, Starc Institute).

**Contraction:** contraction handles deletion using the coherence viewpoint. In particular, this operation assumes that there is a high degree of certainty that the non-responsibility of the *Starc Institute* is only for *Capture 1_7*. Other activities which are still associated with *Starc Institute*, such as *Capture 1_8*, should persist despite the lack of explicit knowledge to support them. In this case we delete only the following:

- (Capture 1_7, carried out by, Starc Institute) // as requested
- (Sequence of shots, carried out by, Starc Institute)
- (Laser scanning acquisition, carried out by, Starc Institute)

Again, implicit facts need not be deleted, so the only actual deletion required is the deletion of (Laser scanning acquisition, carried out by, Starc Institute). For contraction, one should also be careful with the implicit knowledge that is supposed to persist. In our example, the fact that the *Starc Institute* is responsible for *Capture 1_8*, is not explicitly stored and will be lost along with the deletion of (Laser scanning acquisition, carried out by, Starc Institute) unless we explicitly add it back.

We will be referring to the above cases as *actor disassociation* and *actor contraction* respectively and are shown in Fig. 6. Notice that in contrast to disassociation, the contraction operation preserves the association: (Capture 1_8, carried out by, Starc Institute).

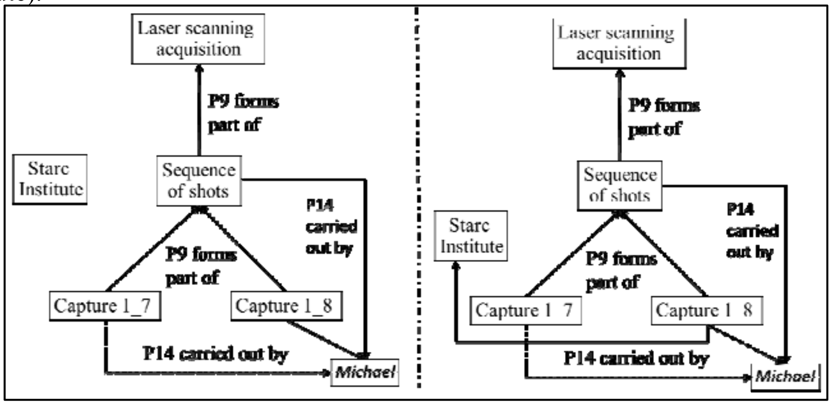

Fig. 6. Actor disassociation (left) and actor contraction (right)

The operation of contraction is very useful in cases of activity delegation, i.e., when specific subactivities are assigned to different actors than their superactivity (e.g., Capture 1_7 has a different actor than the Starc Institute). This assignment can be completed by a combination of a contraction and an addition operation. In this case, Addition and Contraction can be composed to define a "*Replace*" operation. In general, one could compose operations to define more complex, composite ones in various ways, but details on this issue are omitted due to lack of space.

## 4.1 Algorithmic Perspective

The analysis discussed previously can be applied for deriving the exact plan for each change operation involving the rules R1, R2 and R3. Indicatively, we provide below update plans (algorithms) for two operations, namely:

- `DisassociateActorFromActivity (p:Actor, a:Activity)`
- `ContractActorFromActivity (p:Actor, a:Activity)`

```
Algorithm 1: DisassociateActorFromActivity (p:Actor, a:Activity)
1.  if an explicit "carried out by" link exists between a and p then
2.     Remove the requested "carried out by" link between a and p
3.  end if
4.  for each superactivity:superAct of a related to p via the "carried out
    by" link do
5.     Remove possible explicit "carried out by" link between superAct and p
6.  end for
```

```
Algorithm 2: ContractActorFromActivity (p:Actor, a:Activity)
1.  if an explicit "carried out by" link exists between a or a superactivity
    of a and p then
2.     for each maximal subactivity:subAct of a do
3.        Execute AssociateActorToActivity (p, subAct)
4.     end for
5.  end if
6.  if an explicit "carried out by" link exists between a and p then
7.     Remove the requested "carried out by" link between a and p
8.  end if
9.  for each maximal superactivity:supAct of a related to p via the "carried
    out by" link do
10.    for each subactivity:subAct of supAct do
11.       if subAct is not superactivity or subactivity of a then
12.          Add subAct to collection: Col
13.       end if
14.    end for
15. end for
16. Execute DisassociateActorFromActivity (p, a)
17. for each maximal activity:act in Col do
18.    Execute AssociateActorToActivity (p, act)
19. end for
```

Algorithm 1 takes the actor p and activity a in its input, and its purpose is to disassociate p from the responsibility for a. According to the semantics given above, this requires the deletion of all associations of p with all superactivities of a. Note that only explicit links need to be removed, because implicit ones do not actually exist in *K*. Note also that in order to find the superactivities of a, we need to compute (part of) the transitive closure of the "forms part of" property.

Algorithm 2 contracts p from the responsibility for a. This requires, apart from the deletion of all associations of p with all superactivities of a (as in disassociation), the preservation of certain implicit associations that would otherwise be lost In order to avoid adding redundant associations, we add new explicit associations only to the maximal activities in Col or the respective subactivities of a. The complexity of Algorithms 1, 2 is O(NlogN) and O($N^2$) respectively, where N is the number of triples in *K*. These complexities assume that the triples in *K* are originally sorted (in a preprocessing phase); such a sorting costs O(NlogN).

Our operations guarantee that the resulting KB will not contain the deleted triple, either as an explicit or as an implicit fact, given the existing knowledge and the custom inference rules that we consider. In addition, our operations preserve as much as possible of the knowledge in the updated KB under the considered semantics (foundational/coherence for disassociation/contraction respectively). The two observations have been coined as general principles in the belief revision literature [6].

## 5 Related Work

### 5.1 Provenance Storage and Inference Rules

The problem of efficient storage of provenance information has been extensively recognized and studied in the literature. Different methods have been presented for reducing space storage requirements of provenance information. For example, in database operations, provenance minimization via polynomials has been studied in [1]. Another example is [12] in which workflow directed acyclic graphs (DAGs) are transformed into interval encoded tree structures. Furthermore, similar to our notion of property propagation, [5] proposes provenance inheritance methods assuming tree-form models (but not DAGs). However, these are applied on redundant information already stored. Additional techniques have been proposed in [2], where the authors present a more general model than CIDOC CRM and use inference rules for collapsing provenance traces. Although, that model captures mainly only the dependencies among data and thus suffers from overgeneralization and possible loss of information.

Our approach is based on the implicit knowledge that is the logical consequence of the explicit one; note that this is different than tacit knowledge [19], [20] which may be the result of common sense or knowledge. In addition, the reasoning forms that we consider aim at the dynamic completion (deduction) of facts from the original input at query time, rather than at ingestion time and propagate features or properties among entities. This is complementary to reasoning on "data provenance", which traces causal dependencies of individual data elements between input and output. In the latter category we could mention [17], [18]. Lastly, inference rules with annotations are exploited in [4] for scalable reasoning on web data. Although these annotations are indicators of data provenance, they do not directly model it.

### 5.2 Knowledge Evolution in RDF/S

The research field of ontology evolution [9] deals with updating knowledge in ontologies; a detailed survey of the field appears in [8]. However, none of these works considers custom inference rules. Some works (e.g., [22]) address the problem within ontology editors. This is insufficient, because, first, it introduces a huge manual burden to the ontology engineer (who has to manually take care of the complications arising due to inference rules), and, second, it does not consider the standard inference semantics of RDF/S and other ontological languages. Works like [3], [9], [21], [23], [13] have proposed and implemented change semantics which consider the standard inference rules of RDF/S and determine, for each type of change request, the side-effects necessary to properly execute said change taking into account the inference semantics. However, they do not consider custom inference rules or coherence semantics. In certain cases, some flexibility is provided to independently customize or explicitly define the semantics of (some of) the operations [9], [15].

Other works deal with all change operations in a generic manner. For example, [16] proposes a declarative approach which can handle all possible changes on the data part of an RDF/S KB, whereas [14] considers both schema and data changes. These works consider only the standard inference rules of RDF/S (but [14] can be extended to support also custom inference rules) and only the coherence semantics. Our future plans include the generalization of the approach in [14] to support the foundational semantics as well; this will allow us to apply the updates in a generic manner, without having to resort to the specific update plan of each operation.

Finally, we should mention [11] which elaborated on the deletion of triples (including inferred ones) assuming the standard inference rules of RDF/S for both schema and instance update focusing on the "erase" operation. The considered semantics is coherence (only), and the paper shows how one can compute all the "optimal" plans for executing such an erase operation (contraction in our terminology).

## 6 Concluding Remarks

In this paper, we motivated the need for provenance-based inference aiming at the dynamic completion (deduction) of facts from the original input and thus reducing the storage space requirements. Errors in data are only attributed to that input and so the search space for their identification and correction has also been reduced. Furthermore we elaborated on the difficulties of updating a KB using these rules and identified two ways to deal with deletions in this context, based on the philosophical stance against explicit (ingested) knowledge and implicit (inferred) one (foundational and coherence semantics). Based on these ideas, we specified a number of update operations that allow knowledge updating under said inference rules. Although we confined ourselves to CRMdig, and to three specific inference rules, the general ideas behind our work (including the discrimination between foundational and coherence semantics of deletion) can be applied to other models and/or sets of inference rules. A next step is to study the problem in a more generic manner, without resorting to specific, per-operation update plans, in the spirit of [14]. We also plan to conduct experiments to determine the space reduction achieved using our inference rules in real-world

**Acknowledgements.** Work done in the context of NoE APARSEN
(Alliance Permanent Access to the Records of Science in Europe, FP7, Proj.
No 269977).

## 7 References

1. Amsterdamer, Y., Deutch, D., Milo, T., Tannen, V.: On provenance minimization. In: Lenzerini, M., Schwentick, T. (eds.) PODS. pp. 141–152. ACM (2011)
2. Anand, M.K., Bowers, S., McPhillips, T., Ludäscher, B.: Efficient provenance storage over nested data collections. In: Proceedings of EDBT-09, ACM (2009)

3. Bechhofer, S., Horrocks, I., Goble, C., Stevens, R.: Oiled: a reason-able ontology editor for the semantic web. In: Proceedings of KI2001, pp. 396-408 (2001)
4. Bonatti, P.A., Hogan, A., Polleres, A., Sauro, L.: Robust and scalable linked data reasoning incorporating provenance and trust annotations. Web Semantics: Science, Services and Agents on the World Wide Web 9(2) (2011)
5. Chapman, A., Jagadish, H., Ramanan, P.: Efficient provenance storage. In: Proc. of the ACM SIGMOD/PODS Conference.(2008)
6. Dalal, M.: Investigations into a theory of knowledge base revision. In: AAAI. pp.475-479 (1988)
7. Doerr, M.: The cidoc conceptual reference module: An ontological approach to semantic interoperability of metadata. AI Magazine 24(3), 75-92 (2003)
8. Flouris, G., Manakanatas, D., Kondylakis, H., Plexousakis, D., Antoniou, G.: Ontology change: Classification and survey. The Knowledge Engineering Review 23(2), (2008)
9. Gabel, T., Sure, Y., and Völker, J.: D3.1.1.a: Kaon-ontology management infrastructure, SEKT informal deliverable (2004)
10. Gärdenfors, P.: The dynamics of belief systems: Foundations versus coherence theories. Revue Internationale de Philosophie 44, 24-46 (1990)
11. Gutierrez, C., Hurtado, C.A., Vaisman, A.A.: RDFS update: From theory to practice. In: ESWC-11. pp. 93-107 (2011)
12. Heinis, T., Alonso, G.: Efficient lineage tracking for scientific workflows. In: Wang, J.T.L. (ed.) SIGMOD Conference. pp. 1007–1018. ACM (2008)
13. Klein, M., Noy, N.: A component-based framework for ontology evolution. In: Proceedings of the Workshop on Ontologies and Distributed Systems (2003)
14. Konstantinidis, G., Flouris, G., Antoniou, G., Christophides, V.: A formal approach for rdf/s ontology evolution. In: ECAI. pp. 70-74 (2008)
15. Lösch, U., Rudolph, S., Vrandečić, D, Studer, R.: Tempus fugit – towards an ontology update language. In: Proceedings of ESWC-09. pp. 278–292 (2009)
16. Magiridou, M., Sahtouris, S., Christophides, V., Koubarakis, M.: Rul: A declarative update language for rdf. In: Proceedings of ISWC-05 (2005)
17. Moreau, L., Clifford, B., Freire, J., Futrelle, J., Gil, Y., Groth, P.T., Kwasnikowska, N., Miles, S., Missier, P., Myers, J., Plale, B., Simmhan, Y., Stephan, E.G., den Bussche, J.V.: The open provenance model core specification (v1.1). Future Generation Comp. Syst. 27(6), 743-756 (2011)
18. Moreau, L., Missier, P.: The PROV data model and abstract syntax notation. http://www.w3.org/TR/2012/WD-prov-dm-20120202/, February 2012
19. Polanyi, M.: The Tacit Dimension. Doubleday, Garden City, NY (1966)
20. Smith, M.K.: Michael Polanyi and tacit knowledge. The encyclopedia of informal education (2003), www.infed.org/thinkers/polanyi.htm
21. Noy, N., Fergerson, R., Musen, M.: The knowledge model of protégé-2000: Combining interoperability and flexibility. In: Proceedings of EKAW-00 (2000)
22. Stojanovic, L., Motik, B.: Ontology evolution within ontology editors. In: EKAW02/EON Workshop (2002)
23. Sure, Y., Angele, J., Staab, S.: Ontoedit: Multifaceted inferencing for ontology engineering. Journal on Data Semantics 1, pp. 128–152 (2003)
24. Theodoridou, M., Tzitzikas, Y., Doerr, M., Marketakis, Y., Melessanakis, V.: Modeling and querying provenance by extending CIDOC CRM. Distributed and Parallel Databases 27(2), 169-210 (2010)
25. Zeginis, D., Tzitzikas, Y., Christophides, V.: On computing deltas of rdf/s knowledge bases. TWEB 5(3), 14 (2011)