Expressing FactDAG Provenance with PROV-O

✉ Lars Gleim[1], Liam Tirpitz[1], Jan Pennekamp[2], and Stefan Decker[1,3]

[1] Databases and Information Systems, RWTH Aachen University, Germany
[2] Communication and Distributed Systems, RWTH Aachen University, Germany
[3] Fraunhofer FIT, Sankt Augustin, Germany

Abstract. To foster data sharing and reuse across organizational boundaries, provenance tracking is of vital importance for the establishment of trust and accountability, especially in industrial applications, but often neglected due to associated overhead. The abstract FactDAG data interoperability model strives to address this challenge by simplifying the creation of provenance-linked knowledge graphs of revisioned (and thus immutable) resources. However, to date, it lacks a practical provenance implementation.

In this work, we present a concrete alignment of all roles and relations in the FactDAG model to the W3C PROV provenance standard, allowing future software implementations to directly produce standard-compliant provenance information. Maintaining compatibility with existing PROV tooling, an implementation of this mapping will pave the way for practical FactDAG implementations and deployments, improving trust and accountability for Open Data through simplified provenance management.

Keywords: Provenance, Data Lineage, Open Data, Semantic Web Technologies, Ontology Alignment, PROV, RDF, Industry 4.0, Internet of Production, IIoT

1 Introduction

The digital transformation fundamentally changes the utilization of data and knowledge, effectively breaking down traditional isolated knowledge silos [19,16,7]. As enterprises optimize internal and external processes, improve customer interfaces, establish new ecosystems, and develop entirely new business models — such as smart products and services — data sharing, reuse and integration become increasingly important for industry [6,18,15]. To facilitate this transition, the abstract FactDAG data interoperability model [5] was recently introduced. By simplifying the creation of provenance-linked knowledge graphs of revisioned resources, the model enables the integration of data across organizational boundaries, throughout the product life cycles and including both public and private sources [5]. The critical importance of provenance information for data reuse [3] and trust and accountability for derived works has recently been highlighted by the retraction of a prominent study on COVID-19 treatments [12] due to the unverifiable origins of its underlying data. In this paper, we present a concrete alignment of the FactDAG's abstract concepts and relations to the abstract W3C PROV data model [13] and its concrete materialization, the PROV-O ontology [11]. This mapping should serve as a foundation for a practical software implementation of the FactDAG

model, resulting in the simplified standard-compliant creation, management and sharing of data and provenance information. As such, the practical adoption of an implementation of the FactDAG model based on this mapping will ultimately contribute to the reuse of both open and proprietary data, e.g., as envisioned in the Internet of Production [17].

2 FactDAG Provenance

Before detailing the concrete mapping from the FactDAG model to an established W3C provenance model, we first highlight the required foundations of both the FactDAG as well as the relevant PROV standards. Subsequently, we present the contribution of this paper, i.e., our formalized representation.

FactDAG Fundamentals. The previously proposed FactDAG data interoperability model [5] distinguishes the following concepts: *Facts* are globally persistently identified and immutable revisions of arbitrary resources (as in resource-oriented architectures), *Authorities* may be entities (e.g., companies or organizations) that are responsible for those facts, *Processes* describe prototypical interactions with facts, and *ProcessExecutions* refer to their instantiations, introduced to capture individual influences and results (i.e., newly created facts or fact revisions). Additionally, the model uses a single relation (called *influence*, oriented forwards in time) to express relations between the elements of the FactDAG. Thus, the model allows tracing back the origins of facts throughout time, revealing the resources, authorities and processes involved in its conception. For additional details, we refer to the specification of the FactDAG model [5].

Provenance. To maintain compatibility with existing tooling and profit from the extensible nature of the W3C PROV-DM provenance model [13], we embed the FactDAG into a subset of the standardized PROV ontology [11]. The model consists of three fundamental base classes and relations between them: Entity, a physical, digital, or conceptual kind of real or imaginary thing. Activity, something that occurs over a period of time and acts upon or with entities; possibly including consuming, processing, transforming, modifying, relocating, using, or generating entities. Agent, something or someone that bears some form of responsibility for an activity taking place, for the existence of an entity, or for another agent's activity (e.g., an Organization – a subtype of Agent –

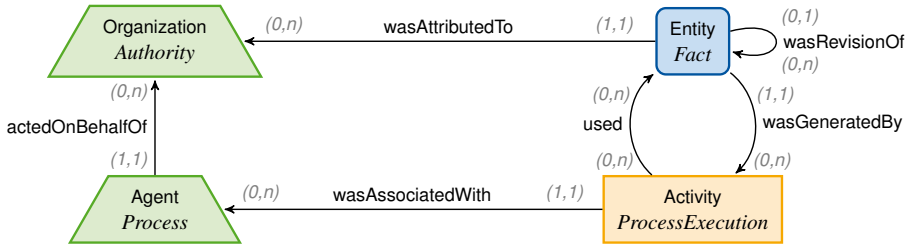

Fig. 1. The elements of the FactDAG model (in italic) and their provenance relations expressed using PROV-O primitives with corresponding *(min,max)*-cardinalities [1]. The shapes represent the PROV core classes Entity, Activity and Agent (with Organization as a subtype), respectively.

for the activities of their employees, or developers for that of their software). We mark concepts and roles from the PROV-O ontology [11] using sans-serif font.

In the following, we propose a formal mapping of the FactDAG concepts into the PROV standard [11] for provenance exchange as illustrated in Figure 1. Finally, we express the relations between elements of the FactDAG using PROV properties and describe the prospective benefits of the presented mapping.

Mapping Types. In the following, we formally define a complete mapping of FactDAG concepts into PROV using description logic [9]:

$$\top_F \equiv \mathit{Fact} \sqcup \mathit{Authority} \sqcup \mathit{Process} \sqcup \mathit{ProcessExecution} \tag{1}$$

$$\mathit{Fact} \sqsubseteq \mathsf{Entity} \tag{2}$$

$$\mathit{ProcessExecution} \sqsubseteq \mathsf{Activity} \tag{3}$$

$$\mathit{Process} \sqsubseteq \mathsf{Entity} \sqcap \mathsf{Agent} \tag{4}$$

$$\mathit{Authority} \sqsubseteq \mathsf{Entity} \sqcap \mathsf{Organization} \tag{5}$$

$$\text{Disjoint}(ProcessExecution, Fact) \tag{6}$$

Eq. (1) defines the set of all FactDAG concepts $\top_F$ which contains all FactDAG elements. All concepts are mapped to the PROV-O serialization of PROV-DM as follows:

Eq. (2): Every *Fact* is a PROV Entity, as we want to track provenance for *Facts* and Entities are the subjects of provenance records in PROV.

Eq. (3): A *ProcessExecution* is modeled as a PROV Activity, since PROV Activities capture interactions with and transformations of Entities, analogously as *ProcessExecutions* do for *Facts*.

Eq. (4): Every *Process* in the FactDAG model is both an Entity (to capture its provenance) as well as an Agent (to model its responsibility for corresponding process executions and interactions with facts).

Eq. (5): Since an *Authority* in the FactDAG model is a company or another organizational unit, it is specifically mapped to the Organization subclass of the PROV Agent, as well as the *Entity* class to enable the tracking of its provenance.

Eq. (6): Finally, the mapping has to satisfy the constraints of the PROV model [21], which notably state, that Entity and Activity are disjoint. Hence, the FactDAG concepts *ProcessExecution* and *Fact* also need to be disjoint.

While this last requirement does not follow directly from the FactDAG model or its requirements, it is needed for compliance with the PROV constraints [21]. However, since it does not limit the general applicability of the FactDAG model, this simple set of definitions provides an adequate mapping of all elements of $\top_F$ to types in PROV.

Relations between Elements of the FactDAG. While the FactDAG model proposes a single *influence* relation oriented *forwards in time* to express the relation between the elements of the FactDAG, fact immutability prevents this information from being added at a later time in practical implementations. Thus, we employ PROV's wasInfluencedBy (which is oriented *backwards in time*) and its more specific subproperties to express influences at fact creation time. Therefore, the provenance relations in the FactDAG are

represented in the following way:

$$\begin{aligned} \textit{Process} \sqsubseteq\ & \geqslant 1\ \mathsf{actedOnBehalfOf.Authority} \\ & \sqcap \leqslant 1\ \mathsf{actedOnBehalfOf.Authority} \end{aligned} \tag{7}$$

$$\begin{aligned} \exists \mathsf{actedOnBehalfOf.}\top_F &\sqsubseteq \textit{Process} \\ \top_F &\sqsubseteq \forall \mathsf{actedOnBehalfOf.Authority} \end{aligned} \tag{8}$$

$$\begin{aligned} \textit{Fact} \sqsubseteq\ & \geqslant 1\ \mathsf{wasAttributedTo.Authority} \\ & \sqcap \leqslant 1\ \mathsf{wasAttributedTo.Authority} \end{aligned} \tag{9}$$

$$\begin{aligned} \exists \mathsf{wasAttributedTo.}\top_F &\sqsubseteq \textit{Fact} \\ \top_F &\sqsubseteq \forall \mathsf{wasAttributedTo.Authority} \end{aligned} \tag{10}$$

$$\begin{aligned} \textit{ProcessExecution} \sqsubseteq\ & \geqslant 1\ \mathsf{associatedWith.Process} \\ & \sqcap \leqslant 1\ \mathsf{associatedWith.Process} \end{aligned} \tag{11}$$

$$\begin{aligned} \exists \mathsf{associatedWith.}\top_F &\sqsubseteq \textit{ProcessExecution} \\ \top_F &\sqsubseteq \forall \mathsf{associatedWith.Process} \end{aligned} \tag{12}$$

$$\begin{aligned} \exists \mathsf{used.}\top_F &\sqsubseteq \textit{ProcessExecution} \\ \top_F &\sqsubseteq \forall \mathsf{used.Fact} \end{aligned} \tag{13}$$

$$\begin{aligned} \textit{Fact} \sqsubseteq\ & \geqslant 1\ \mathsf{wasGeneratedBy.ProcessExecution} \\ & \sqcap \leqslant 1\ \mathsf{wasGeneratedBy.ProcessExecution} \end{aligned} \tag{14}$$

$$\begin{aligned} \exists \mathsf{generatedBy.}\top_F &\sqsubseteq \textit{Fact} \\ \top_F &\sqsubseteq \forall \mathsf{generatedBy.ProcessExecution} \end{aligned} \tag{15}$$

$$\begin{aligned} \exists \mathsf{wasRevisionOf.}\top_F &\sqsubseteq \textit{Fact} \\ \top_F &\sqsubseteq \forall \mathsf{wasRevisionOf.Fact} \\ \textit{Fact} &\sqsubseteq \leqslant 1 \mathsf{wasRevisionOf.Fact} \end{aligned} \tag{16}$$

Intuitively, these statements express the following relations and restrictions, as illustrated in Figure 1 with their corresponding cardinality restrictions.

Eq. (7): A *Process* always actedOnBehalfOf exactly one *Authority*.

Eq. (8): Between FactDAG elements, the actedOnBehalfOf property is only used to relate *Processes* to *Authorities*.

Eq. (9): All *Facts* are attributedTo exactly one responsible *Authority*.

Eq. (10): Between FactDAG elements, the wasAttributedTo property is only used to relate *Facts* to *Authorities*.

Notably, Eq. 7 and Eq. 9 entail that neither a *Fact*, nor a *Process* may be directly shared among *Authorities*. Instead, copies of the same fact may be related across authorities through the PROV *wasRevisionOf* or alternateOf predicates.

Eq. (11): A *ProcessExecution* is always associatedWith exactly one *Process*.

Eq. (12): Between FactDAG elements, the associatedWith property always links from a *ProcessExecution* to a *Process*.

Eq. (13): *Facts* that impacted a *ProcessExecution* are recorded using the used relation. A *ProcessExecution* may have *used* arbitrary many *Facts* and used exclusively relates *ProcessExecutions* to *Facts*.

Eq. (14): If a *ProcessExecution* results in new *Facts*, these facts are linked to the process execution which generated them with wasGeneratedBy. Therefore every *Fact* has exactly one wasGeneratedBy relation pointing to the *ProcessExecution* that generated it.

Eq. (15): Between FactDAG elements, the wasGeneratedBy property is only used to link from *Facts* to *ProcessExecutions*.

Eq. (16): If a *Fact* is a new revision of a previously existing resource, it is linked to its direct predecessor using wasRevisionOf. Therefore, between FactDAG elements, the wasRevisionOf always links from one *Fact* to another *Fact*, and every *Fact* is the subject of at most one wasRevisionOf relation.

Overall, the union of the properties wasAttributedTo, wasRevisionOf, wasGeneratedBy, actedOnBehalfOf, used and wasAssociatedWith (which are all sub-properties of wasInfluencedBy) encompasses all instances of the *was influenced by* relation which is used instead of the *influenced* relation of the FactDAG. Thus, all cases of the FactDAG influence relation are mapped to an appropriate property in PROV, providing standard-compliant provenance semantics for the FactDAG model. Subsequently, software implementations of the FactDAG model may directly generate provenance information in compatibility with the W3C PROV model and the advanced tooling already existing for end users to leverage it [8,10,14]. As such, the presented mapping has the potential to contribute towards improved transparency, trust and accountability for Open Data and practical data reuse.

3 Conclusion

In this work, we provide a mapping of all types and their relations from the FactDAG model to a subset of the W3C recommended PROV data model for provenance information. Therefore, the semantics of the FactDAG model can be fully expressed with PROV, providing a standardized framework for provenance exchange, allowing for the reuse of tools developed for PROV, such as reasoning within the PROV semantics.

In conjunction with suitable persistent identifiers, such as timestamped URLs [4] for resource identification, our work thus provides a practical basis for software implementations of the FactDAG data interoperability model to generate standard-compliant provenance information and thus to further simplified provenance management for Open Data and data reuse. It thus paves the way for applications such as process mining techniques [22], provenance-based process planning [20] and extensions towards process plan modeling, e.g. using the P-Plan extension [2] of the PROV standard, in future work. Finally, the presented approach may facilitate further adoption of Semantic Web Technologies in industry and support Open Data sharing and reuse through establishing interorganizational knowledge graphs of provenance-linked resources, as proposed by the FactDAG model.

Acknowledgments. Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC-2023 Internet of Production – 390621612.

References

1. Abrial, J.: Data Semantics. In: IFIP Working Conference on Data Base Management (1974)
2. Garijo, D., Gil, Y.: Augmenting PROV with Plans in P-PLAN: Scientific Processes as Linked Data. In: LISC @ ISWC (2012)
3. Gleim, L., Decker, S.: Open Challenges for the Management and Preservation of Evolving Data on the Web. In: MEPDaW @ ISWC (2020)
4. Gleim, L., Decker, S.: Timestamped URLs as Persistent Identifiers. In: MEPDaW @ ISWC (2020)
5. Gleim, L., Pennekamp, J., Liebenberg, M., Buchsbaum, M., Niemietz, P., Knape, S., Epple, A., Storms, S., Trauth, D., Bergs, T., Brecher, C., Decker, S., Lakemeyer, G., Wehrle, K.: FactDAG: Formalizing Data Interoperability in an Internet of Production. IEEE Internet of Things J. **7**(4) (2020)
6. Ibarra, D., Ganzarain, J., Igartua, J.I.: Business model innovation through Industry 4.0: A review. Procedia Manufacturing **22**, 4–10 (2018)
7. Jarke, M.: Data Sovereignty and the Internet of Production. In: CAiSE (2020)
8. Kohwalter, T., Oliveira, T., Freire, J., Clua, E., Murta, L.: Prov viewer: A graph-based visualization tool for interactive exploration of provenance data. In: IPAW (2016)
9. Krötzsch, M., Simancik, F., Horrocks, I.: A Description Logic Primer. arXiv:1201.4089 (2012)
10. Luger, S., Stitz, H., Gratzl, S., Gehlenborg, N., Streit, M.: Interactive visualization of provenance graphs for reproducible biomedical research. In: IEEE InfoVis – Posters (2015)
11. McGuinness, D., Sahoo, S., Lebo, T.: PROV-O: The PROV Ontology. W3C rec. (2013)
12. Mehra, M.R., Desai, S.S., Ruschitzka, F., Patel, A.N.: RETRACTED: Hydroxychloroquine or chloroquine with or without a macrolide for treatment of COVID-19: a multinational registry analysis. The Lancet (2020)
13. Missier, P., Moreau, L.: PROV-DM: The PROV Data Model. W3C rec. (2013)
14. Moreau, L., Batlajery, B.V., Huynh, T.D., Michaelides, D., Packer, H.: A templating system to generate provenance. IEEE Trans. Softw. Eng. (2017)
15. Pennekamp, J., Buchholz, E., Lockner, Y., Dahlmanns, M., Xi, T., Fey, M., Brecher, C., Hopmann, C., Wehrle, K.: Privacy-Preserving Production Process Parameter Exchange. In: ACSAC (2020)
16. Pennekamp, J., Dahlmanns, M., Gleim, L., Decker, S., Wehrle, K.: Security Considerations for Collaborations in an Industrial IoT-based Lab of Labs. In: IEEE GCIoT (2019)
17. Pennekamp, J., Glebke, R., Henze, M., Meisen, T., Quix, C., Hai, R., Gleim, L., Niemietz, P., Rudack, M., Knape, S., Epple, A., Trauth, D., Vroomen, U., Bergs, T., Brecher, C., Bühring-Polaczek, A., Jarke, M., Wehrle, K.: Towards an Infrastructure Enabling the Internet of Production. In: IEEE ICPS (2019)
18. Pennekamp, J., Henze, M., Schmidt, S., Niemietz, P., Fey, M., Trauth, D., Bergs, T., Brecher, C., Wehrle, K.: Dataflow Challenges in an Internet of Production: A Security & Privacy Perspective. In: ACM CPS-SPC (2019)
19. Schrauf, S., Berttram, P.: Industry 4.0: How digitization makes the supply chain more efficient, agile, and customer-focused. Strategy& (2016)
20. Silva, M.F., Baião, F.A., Revoredo, K.: Towards planning scientific experiments through declarative model discovery in provenance data. In: IEEE eScience (2014)
21. Tom De Nies: Constraints of the PROV Data Model (2013)
22. Zeng, R., He, X., Li, J., Liu, Z., van der Aalst, W.M.: A method to build and analyze scientific workflows from provenance through process mining. In: TaPP (2011)