Towards a graphical tool for modeling scientific workflows' provenance according to the W3C PROV standard

Marcos Alves Vieira[1,2], **Sergio T. Carvalho**[2]

[1]Instituto Federal de Educação, Ciência e Tecnologia Goiano (IF Goiano)
Iporá – GO – Brasil

[2]Instituto de Informática – Universidade Federal de Goiás (UFG)
Goiânia – GO – Brasil

marcos.vieira@ifgoiano.edu.br, sergiocarvalho@ufg.br

Abstract. *Provenance makes it possible to describe information about the steps involved in the production of a piece of data and allows an assessment of its quality, reliability, or credibility. When it comes to scientific workflows, provenance establishes the relationships between the artifacts associated with a given set of simulations and can be used to enable: (i) their sharing with the scientific community, (ii) the reproducibility of the results, or (iii) the evaluation of erroneous outputs. This paper presents the work in progress towards building a graphical provenance modeling tool conforming to the W3C PROV standard and following Model-Driven Engineering (MDE) concepts. The modeling tool can be used to model scientific workflows' provenance, enabling, for instance, their visual representation, reproducibility, and sharing.*

1. Introduction

Scientific workflows are abstractions representing a set of connected activities, used to support the modeling and description of scientific experiments [Sembay et al. 2020]. Provenance provides "information about entities, activities, and people involved in producing a piece of data or thing, which can be used to form assessments about its quality, reliability, or trustworthiness" [Groth and Moreau 2013]. In the context of scientific workflows, provenance allows the representation of the history of an experiment and is an essential component to enable the reproducibility of results, the sharing and reuse of knowledge in the scientific community, both for the data produced by scientific workflows and for its specification [Davidson and Freire 2018].

Provenance establishes the relationships between the artifacts associated with a given set of simulations, including the software execution, the interrelationships between different software versions, the simulation parameters, and the input and output data. By providing data and software provenance, scientists can not only determine accurate experimental conditions from previous simulations, thus increasing confidence in the results, but also detect when outputs may be erroneous due to bugs [Conquest and Stiber 2021].

However, the available graphical tool options for modeling scientific workflows do not handle provenance aspects using an interoperable provenance description standard, which would facilitate automated construction of provenance models by software (*e.g.*, for simulation of scientific workflows) or the instantiation of these models by software.

Therefore, this paper presents the ongoing work towards building a graphical provenance modeling tool, which can be used to model scientific workflows' provenance, enabling, for instance, their documentation, visual representation, sharing, and reproducibility. The graphical modeling tool aims to allow the production of provenance models according to the W3C PROV standard and is being developed using Model-Driven Engineering (MDE) techniques, based on the Meta-Object Facility (MOF) metamodeling architecture, using the Eclipse Modeling Framework (EMF) and Eclipse Sirius.

The remainder of this paper is organized as follows: Section 2 presents the theoretical and technological basis to support developing the modeling tool; in Section 3 there is an analysis of the related works and a brief discussion relating them to our proposal; Section 4 reports on the current development stage of the graphical provenance modeling tool; finally, in Section 5 the final considerations are outlined.

2. Theoretical and technological background

This section brings the theoretical and technological concepts required to develop the graphical provenance modeling tool.

2.1. Theoretical background

In general, the term "provenance" refers to any information that describes a production process of a product, for example, a piece of data or a physical object [Herschel et al. 2017]. The W3C Provenance Working Group defines provenance as "information about entities, activities, and people involved in producing a piece of data or thing, which can be used to form assessments about its quality, reliability, or trustworthiness." The term originates from the Latin word "*prōvenīre*", which means "coming from". Data provenance provides the history of the origins of all changes to an object, the list of components that have forwarded or processed that object, and the users who have viewed or changed it [Liang et al. 2017]. Data provenance represents the history of an experiment and provides essential background for reproducing and interpreting the results obtained from running the experiment [Alves et al. 2020]. The term "data provenance" refers to mechanisms and techniques for obtaining and recording information about the origin of data and the transformations that have occurred in data to its current state. A collection of this information can be called "provenance data" or "provenance information".

The concept of Model-Driven Engineering (MDE) considers that models are the main artifacts in the development of a system. According to this approach, models do not only serve to describe or document a software, but also to act on its development, maintenance, and operation [Schmidt 2006]. A model is a high-level graphic or textual representation of a system, where each of its elements is a virtual representation of a component present in the real system. The relationships and abstractions used in a model are described by a metamodel [Völter et al. 2013].

Given the models' popularity, came the need for standardization for the construction of metamodels and models. Thus, the Object Management Group (OMG)[1] presented a four-layer metamodel architecture, called Meta-Object Facility (MOF), where each element of a lower layer is an instance of an upper layer's element. The MOF layers can be described as follows:

[1]http://www.omg.org

- **M3 layer**: represents the MOF meta-metamodel, also called MOF Model, used to create metamodels. The MOF model formalizes its abstractions, eliminating the need for a higher level. Another example of a member of this layer is Ecore, which is based on MOF. In Section 4, we present the conversion of the W3C PROV standard data model, called PROV-DM, to an instance of the Ecore meta-metamodel, using Eclipse EMF.
- **M2 layer**: contains the metamodels that can be used to model specific domain systems. The Unified Modeling Language (UML), the HyperText Markup Language (HTML) and the metamodel derived from PROV-DM, presented in Section 4, are member examples of this layer.
- **M1 layer**: composed of models that describe systems using the definitions contained in their respective metamodels present in M2.
- **M0 layer**: contains the entities or objects that form the runtime system, which is created from the definitions present in M1.

A metamodel can be considered a Domain-Specific Modeling Language (DSML), which is "a textual or graphical language that provides, through appropriate notations and abstractions, expressive power with a focus on a particular problem domain, to visualize, specify, construct, and document artifacts of a software system" [Chiprianov et al. 2014]. Similar to any language, DSMLs have two main components [Ferreira Filho 2014]: syntax and semantics. The syntax of a DSML can be divided into abstract syntax and concrete syntax. The abstract syntax defines its concepts and the relationships between them, while the concrete syntax maps these concepts into visual elements that are used in models. The semantics of a DSML is the meaning of the syntax representations.

2.2. Technological background

The W3C PROV is a set of documents that define many aspects necessary to enable the interoperable exchange of provenance information in heterogeneous environments. Proposed by the W3C's Provenance Working Group [Gil et al. 2013], PROV enables to represent provenance in the shape of three main concepts: *Entity*, *Agent*, *Activity*. PROV-DM [Moreau et al. 2013] is the main component of the W3C PROV family and establishes the core structures of the PROV standard by defining a data model for provenance.

Figure 1 shows the associations between the elements involved in a provenance record in W3C PROV. An *Entity* is an object on which the provenance record is being made and can be a physical or digital object. An *Activity* describes the changes occurring to an Entity, for example, if a D_t document is generated by translating a D document, then the Activity in this example would be the translation itself. Finally, an *Agent* is the performer of an Activity, which can be a person, software, an inanimate object, or a company, among others [Gil et al. 2013].

In order to enable serialization and storage of instances of the W3C PROV model, its working group also provides: *(i)* an XML schema, called PROV-XML; *(ii)* a concrete syntax in the form of the PROV-N language, allowing provenance records to be represented in a compact, human-readable manner; and *(iii)* the PROV-JSON, which allows provenance representation in the JavaScript Object Notation (JSON) format.

The Eclipse Modeling Framework (EMF) [Steinberg et al. 2008] is a modeling framework built on top of the Eclipse Integrated Development Environment (IDE). EMF

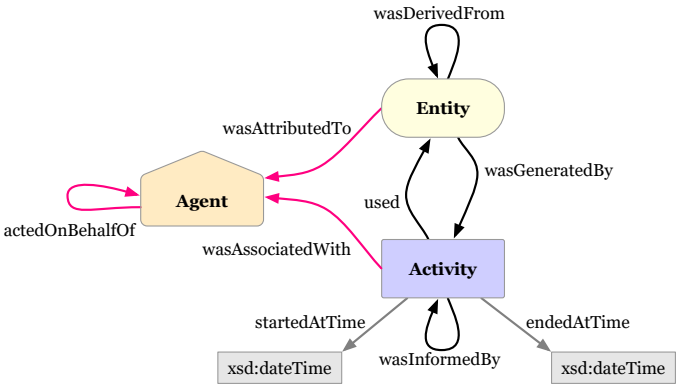

Figure 1. Core W3C PROV data model components [Lebo et al. 2013].

provides mechanisms for the creation, editing, and validation of models and metamodels, in addition to allowing the generation of code from the models. For such, EMF allows the generation of an implementation in Java language, so that each one of the classes of the metamodel (called metaclasses) corresponds to a Java class. This way, these classes can be instantiated to create models in accordance with the metamodel. The EMF also allows creating editors for models in accordance with their metamodels. The metamodels built in EMF are instances of the Ecore meta-metamodel, which, in turn, is based on the MOF meta-metamodel. Thus, Ecore is the central language of EMF [Steinberg et al. 2008].

Sirius[2] is an Eclipse project that facilitates the creation of a graphical modeling tool using Eclipse modeling technologies, including EMF and Graphical Modeling Framework (GMF). Sirius directly interprets the graphical editor model. The ability to interpret the model enables immediate visualization of the resulting editor while modeling. This significantly speeds up the development cycles of a modeling tool [Jäger et al. 2016]. The effort for software development is reduced employing these tools, and the developer can better focus on the activity of system modeling [Jäger et al. 2016].

3. Related work

Thanks to the W3C PROV standardization, many simulation platforms are considering its adoption to store and manage their provenance. One example is the work of [Pignotti et al. 2013], where the authors propose an approach for representing and querying the provenance of agent-based simulations. Their work uses PROV-DM to represent the simulation provenance. Specifically, in their model, entities represent the simulation source code, data, input/output parameters, library, or compiler version. An agent can be a user, an operating system, a particular hardware component, and a software tool. An activity corresponds to a specific action, such as design, data collection, adjustment, and verification. A relationship between these elements can be *wasGeneratedBy*, *used*, *wasAttributedTo*, *wasInformedBy*, or *wasDerivedFrom*.

VisTrails [Silva et al. 2010] is an open source distributed scientific workflow management and provenance system that provides support for simulations, data exploration and visualization. The data provenance, the workflows that originate this data, and their execution are persisted as XML files or in a relational database. VisTrails is available for

[2]https://www.eclipse.org/sirius/

Windows, Linux and Mac computers, having its first version released in October 2007 and its latest stable version is available for download[3] since May 2016.

Another application example of the W3C PROV standard comes from the work of [Suh and Ma 2017]. The authors adopt PROV to collect and standardize simulations provenance. For a given simulation, it is possible to obtain a JSON file, representing its provenance in the W3C PROV standard.

The main difference between the works mentioned in this section and our proposal lies in the use of the W3C PROV standard together with a graphical tool to enable the development of provenance models compliant with this standard.

4. Graphical tool for modeling W3C PROV provenance models

In [Vieira and Carvalho 2020] we present a process for creating graphical tools for building models that conform to an EMF metamodel. As an example, we demonstrated the creation of a modeling tool for the ubiquitous computing domain, allowing modeling of smart spaces and their constituents. At the time, the Eugenia tool and other languages of the Epsilon[4] family were used to support building the graphical tool. However, Eugenia produces modeling tools using the Graphical Modeling Framework (GMF) Tooling, which is no longer actively maintained by the Eclipse community. Therefore, to ensure greater maintainability, we chose to use Eclipse Sirius to build the graphical tool described in this paper, as suggested in the current Eugenia documentation.

For the graphical provenance modeling tool development, two main steps were established: *(i)* convert the PROV-DM model to an EMF metamodel, and *(ii)* build the graphical modeling tool using Eclipse Sirius. Step 1 consisted of converting the PROV-DM model into a single, cohesive EMF metamodel, which was conceived following the descriptions of the W3C PROV components presented in [Moreau et al. 2013]. The metamodel was tested by instantiating provenance records found in the literature (*e.g.* [Moreau and Groth 2013]). This step is completed and the abstract syntax of the resulting metamodel can be seen in Figure 2.

The semantics of the metamodel is defined by relationships and their multiplicities, represented by edges and their numbering, respectively. The metaclasses in blue color are part of the core model: *Entity*, *Activity*, and *Agent*. To simplify the explanation of the PROV-DM model, its creators categorized the elements into six components, each one grouping members for specific purposes. The metaclasses in yellow color correspond to Component 1 of the model and represent some interrelationships between an entity and an activity: *Used* (Usage), *WasGeneratedBy* (Generation), *WasStartedBy* (Start), *WasEndedBy* (End), *WasInvalidatedBy* (Invalidation), and *WasInformedBy* (Communication). The metaclasses in red color depict Component 2 and denote the derivation of an entity by an activity from another entity, represented by the *WasDerivedFrom* metaclass and its subtypes: *WasRevisionOf* (Revision), *WasQuotedFrom* (Quotation), and *HasPrimarySource* (Primary Source). The metaclasses in green color constitute Component 3 and represent the relationships *WasAttributedTo* (Attribution), *WasAssociatedWith* (Association), and *ActedOnBehalfOf* (Delegation), relating agents to entities, activities,

[3]https://www.vistrails.org/index.php/Downloads

[4]https://www.eclipse.org/epsilon/

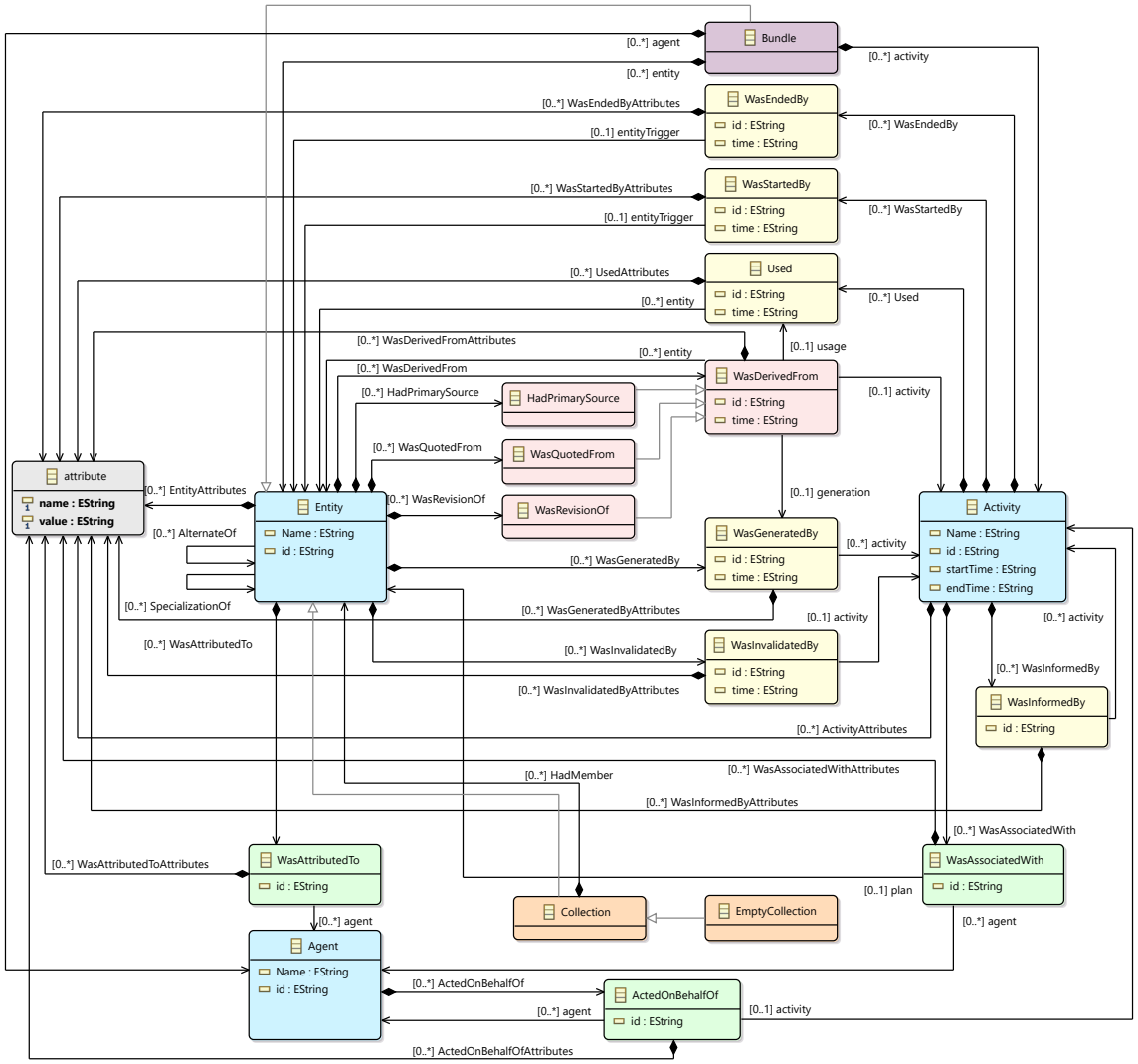

Figure 2. EMF metamodel representing the W3C PROV data model (PROV-DM).

and agents, respectively. Component 4, represented by the metaclass *Bundle*, is concerned with bundles, a mechanism to support provenance of provenance. Component 5 is denoted by the relations *SpecializationOf* (Specialization) and *AlternateOf* (Alternate) between entities. Finally, Component 6 concerns collections of entities, which in turn are also entities, and are represented by the orange color metaclasses *Collection* and *EmptyCollection*.

Step 2 consists of developing the graphical tool using Eclipse Sirius and is currently under development. To this end, the following are defined: *(i)* the implementation of the metamodel's concrete syntax (*i.e.*, graphical representation) using the diagram perspective, following the style definitions[5] and *(ii)* the implementation of the W3C PROV standard constraint rules[6].

Once completed, the modeling tool will enable the graphical creation of provenance models that conform to the W3C PROV standard and following its stylization ru-

[5]https://www.w3.org/2011/prov/wiki/Diagrams
[6]http://www.w3.org/TR/prov-constraints/

les, such as the provenance model presented by Figure 3. These models can be used to understand how transformations occur in the data managed by a system, thus being the first step in supporting data provenance. Once data provenance is modeled, it needs to be captured, structured, and stored in a repository so that it can be queried and then taken advantage of its benefits. The modeling tool introduced in this paper aims to enable the modeling of data provenance, being the other related aforementioned activities beyond the scope of this work.

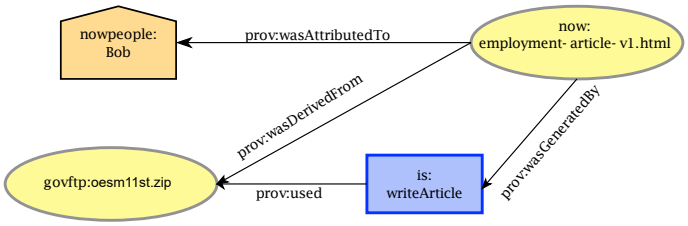

Figure 3. Example of a provenance model representing how a journalistic article was written [Moreau and Groth 2013].

5. Concluding remarks

Provenance in the context of scientific workflows establishes the relationships between the artifacts associated with a given set of simulations and enables the reproducibility of results, the sharing and reuse of knowledge in the scientific community. However, the graphical tool options for modeling scientific workflows do not address provenance aspects using an interoperable provenance description standard.

In this paper, we present the ongoing work towards building a graphical provenance modeling tool that can be used to model scientific workflows' provenance, enabling, for example, their documentation, visual representation, sharing and reproducibility. The graphical tool is being developed using Model-Driven Engineering (MDE) techniques and aims to produce provenance models following the W3C PROV standard. The next steps of this work consist of finishing the graphical modeling tool and evaluating it with experts in the system modeling field. Future work includes implementing model-to-text conversion techniques, enabling automated conversion of the graphical models created using the tool into textual models, in accordance with the PROV-N language.

References

[Alves et al. 2020] Alves, R., Frota, Y., and de Oliveira, D. (2020). Gerência de dados de proveniência distribuídos de experimentos científicos: um mapeamento sistemático. In *Anais do XIV Brazilian e-Science Workshop*, pages 97–104, Porto Alegre, RS, Brasil. SBC.

[Chiprianov et al. 2014] Chiprianov, V., Kermarrec, Y., Rouvrais, S., and Simonin, J. (2014). Extending enterprise architecture modeling languages for domain specificity and collaboration. *Software & Systems Modeling*, 13(3):963–974.

[Conquest and Stiber 2021] Conquest, J. and Stiber, M. (2021). Software and Data Provenance as a Basis for eScience Workflow. In *2021 IEEE 17th International Conference on eScience*.

[Davidson and Freire 2018] Davidson, S. B. and Freire, J. (2018). Provenance and scientific workflows: Challenges and opportunities. In *Proceedings of the 2018 ACM SIGMOD International Conference on Management of Data*, SIGMOD '18, page 1345–1350, New York, NY, USA. Association for Computing Machinery.

[Ferreira Filho 2014] Ferreira Filho, J. B. (2014). *Leveraging model-based product lines for systems engineering*. PhD thesis, Université Rennes 1, Paris, France.

[Gil et al. 2013] Gil, Y., Miles, S., Belhajjame, K., Deus, H., Garijo, D., Klyne, G., Missier, P., Soiland-Reyes, S., and Zednik, S. (2013). PROV Model Primer. Available online: https://www.w3.org/TR/2013/NOTE-prov-primer-20130430/.

[Groth and Moreau 2013] Groth, P. and Moreau, L. (2013). PROV-Overview. An Overview of the PROV Family of Documents. Project report, World Wide Web Consortium.

[Herschel et al. 2017] Herschel, M., Diestelkämper, R., and Ben Lahmar, H. (2017). A survey on provenance: What for? What form? What from? *The VLDB Journal*, 26(6):881–906.

[Jäger et al. 2016] Jäger, S., Maschotta, R., Jungebloud, T., Wichmann, A., and Zimmermann, A. (2016). Creation of domain-specific languages for executable system models with the eclipse modeling project. In *2016 Annual IEEE Systems Conference (SysCon)*, pages 1–8.

[Lebo et al. 2013] Lebo, T., Sahoo, S., McGuinness, D., Belhajjame, K., Cheney, J., Corsar, D., Garijo, D., Soiland-Reyes, S., Zednik, S., and Zhao, J. (2013). PROV-O: The PROV Ontology. Available online: https://www.w3.org/TR/2013/REC-prov-o-20130430/.

[Liang et al. 2017] Liang, X., Shetty, S., Tosh, D., Kamhoua, C., Kwiat, K., and Njilla, L. (2017). ProvChain: A Blockchain-Based Data Provenance Architecture in Cloud Environment with Enhanced Privacy and Availability. In *2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)*, pages 468–477. IEEE.

[Moreau and Groth 2013] Moreau, L. and Groth, P. (2013). *Provenance – An Introduction to PROV*. Synthesis Lectures on the Semantic Web: Theory and Technology. Morgan and Claypool Life Sciences, San Rafael, CA.

[Moreau et al. 2013] Moreau, L., Missier, P., Belhajjame, K., B'Far, R., Cheney, J., Coppens, S., Cresswell, S., Gil, Y., Groth, P., Lebo, G. K. T., McCusker, J., Miles, S., Myers, J., and Sahoo, S. (2013). PROV-DM: The PROV Data Model. Available online: https://www.w3.org/TR/2013/REC-prov-dm-20130430/.

[Pignotti et al. 2013] Pignotti, E., Polhill, G., and Edwards, P. (2013). Using provenance to analyse agent-based simulations. In *Proceedings of the Joint EDBT/ICDT 2013 Workshops*, EDBT '13, page 319–322, New York, NY, USA. Association for Computing Machinery.

[Schmidt 2006] Schmidt, D. C. (2006). Guest editor's introduction: Model-driven engineering. *Computer*, 39(2):0025–31.

[Sembay et al. 2020] Sembay, M. J., de Macedo, D. D. J., and Lima Dutra, M. (2020). A method for collecting provenance data: A case study in a brazilian hemotherapy center. In Mugnaini, R., editor, *Data and Information in Online Environments*, pages 89–102, Cham. Springer.

[Silva et al. 2010] Silva, C. T., Anderson, E., Santos, E., and Freire, J. (2010). Using VisTrails and provenance for teaching scientific visualization. *Computer Graphics Forum*, 30(1):75–84.

[Steinberg et al. 2008] Steinberg, D., Budinsky, F., Merks, E., and Paternostro, M. (2008). *EMF: Eclipse Modeling Framework*. Pearson Education.

[Suh and Ma 2017] Suh, Y.-K. and Ma, J. (2017). Superman: A novel system for storing and retrieving scientific-simulation provenance for efficient job executions on computing clusters. In *2017 IEEE 2nd International Workshops on Foundations and Applications of Self* Systems (FAS*W)*, pages 283–288.

[Vieira and Carvalho 2020] Vieira, M. A. and Carvalho, S. T. (2020). *Building Models for Ubiquitous Application Development in a Model-Driven Engineering Approach*, pages 115–147. Springer International Publishing, Cham.

[Völter et al. 2013] Völter, M., Stahl, T., Bettin, J., Haase, A., and Helsen, S. (2013). *Model-driven software development: technology, engineering, management*. John Wiley & Sons.