# Model-based Construction of Enterprise Architecture Knowledge Graphs

Philipp-Lorenz Glaser, Syed Juned Ali, Emanuel Sallinger and Dominik Bork

To appear in:

*Enterprise Design, Operations and Computing, 26th International Conference, EDOC 2022, Bolzano, Italy, October 3-7, 2022, Proceedings*

# Model-based Construction of Enterprise Architecture Knowledge Graphs

Philipp-Lorenz Glaser[1 [0000-0002-0710-8052]], Syed Juned Ali[1 [0000-0003-1221-0278]], Emanuel Sallinger[2 [0000-0001-7441-129X]], and Dominik Bork[1] ✉[0000-0001-8259-2297]

[1]Business Informatics Group, TU Wien, Austria
[2]Database and Artificial Intelligence Group, TU Wien, Austria
{philipp-lorenz.glaser,syed.juned.ali,emanuel.sallinger,dominik.bork}@tuwien.ac.at

**Abstract.** Enterprise Architecture offers guidelines for the coherent, model-based design and management of enterprises. EA models provide a layered, integrated, and cohesive representation of the enterprise, enabling communication, analysis, and decision making. With the increasing size of EA models, automated analysis becomes essential. However, advanced model analysis is neither incorporated in current EA methods like ArchiMate nor supported by existing EA tools like Archi. Knowledge Graphs (KGs) can effectively organize and represent knowledge and enable reasoning to utilize this knowledge, e.g., for decision support. This paper introduces a model-based Enterprise Architecture Knowledge Graph (EAKG) construction method and shows how starting from ArchiMate models, an initially derived EAKG can be further enriched by EA-specific and graph characteristics-based knowledge. The introduced EAKG entails new representation and reasoning methods applicable to EA knowledge. As a proof of concept, we present the results of a first Design Science Research Cycle aiming to realize an Archi plugin for the EAKG that enables analysis of EA Smells within ArchiMate models.

**Keywords:** Enterprise Architecture · Knowledge Graph · Modeling Tool · ArchiMate · Archi

## 1 Introduction

The transformation of information systems triggered enterprises to adopt enterprise architecture (EA) as a means to manage the complexity and evolution of the enterprise [8]. EA enables comprehensive management and decision-making based on the models of the organization. An enterprise is typically described through multiple EA layers such as *Business*, *Application*, and *Technology*. EA models are graphical representations that provide valuable support, e.g., integrated IT and business decision-making [12], planning future states of the enterprise, and improving the business and IT alignment [19]. To support all these functions, EA models need to be analyzed efficiently. Such EA analysis involves querying models to evaluate various properties [38]. However, holistic EA models grow in size and complexity, thereby hampering manual human analysis, while advanced and automated analysis of EA models is surprisingly underrepresented in research and EA tooling so far [50].

EA modeling tools do not take full advantage of the several structural properties of EA models represented as graphs, such as the differentiation of relations between elements, the discovery of paths, clusters, or graph metrics. Current approaches are often tied to a concrete EA approach, offering a limited set of visualization techniques. EA modeling tools offer different features based on the supported EA approach and the analytical capabilities provided and thus restrict the kind of analysis they support [37]. The need for proper tool support was pointed out in the past as one EA [47] and business information systems modeling [21] research gap. We instantiate this gap in the following by substantiating a need for a generic and advanced EA analysis tool that utilizes the full potential of the graphical structure of EA models.

Knowledge Graphs (KG) represent interlinked descriptions of entities – objects, events, and concepts. Recently, the use of KGs in conceptual modeling, model-driven software engineering, and EA has been explored (cf., [11, 34, 44, 46, 49]). KGs can organize and represent knowledge to ease advanced reasoning (e.g., rule-based and machine learning-based) [15] and to provide question answering, recommendation, and information retrieval solutions [54].

In the context of EA, graph-based formalisms have been applied for the representation and reasoning of EA models [46, 50]. However, these works are merely constrained to the explicit knowledge encoded by the EA model (i.e., no further knowledge enrichment) and basic model analysis (i.e., no KG reasoning). We propose the model-based construction and enrichment of Enterprise Architecture Knowledge Graphs (EAKGs) to exploit the benefits of KG-based representation and reasoning in EA. EAKGs enable AI-based applications for EA model analysis. We further report on developing an EAKG plugin for the Archi toolkit. The plugin visualizes and analyses the EAKG and supports the EAKG knowledge enrichment. The EAKG provides a generic and unified intermediary representation of EAs, making our approach easily extensible for integrating other graph-based EA analysis tools. Our main contributions thus include (*i*) model-based construction and enrichment of EAKG, (*ii*) development of an Archi plugin for analysis and visualization of the EA models, and (*iii*) feasibility evaluation using a case-based approach.

This work reports Design Science Research (DSR) [27]. In particular, we build and evaluate the EAKG plugin for Archi that implements our conceptual contribution, the *model-based construction of EAKGs*.

In the remainder of this paper, first, Section 2 presents the relevant backgrounds and related works on EA Management, KGs, and their combination. We propose an approach for model-based construction of EAKGs in Section 3. The developed EAKG Archi plugin is presented in Section 4. Section 5 reports the results of a case-based evaluation before we finally providing a conclusive discussion of this paper in Section 6.

## 2 Background

Enterprise Architecture Management (EAM) is a "*management practice that establishes, maintains and uses a coherent set of guidelines, architecture principles and governance regimes that provide direction for and practical help with the design and the development of an enterprise's architecture in order to achieve its vision and strategy*" [1]. The most used modeling language, standardized by the Open Group, is ArchiMate [32, 39]. ArchiMate adopts a layered view of an enterprise depicted by the ArchiMate Framework, where the core entities of an enterprise are categorized along *layers* and *aspects*. A strength of ArchiMate is the ability to cover relevant aspects of an enterprise in a holistic and integrated manner. Shortcomings of ArchiMate are its limited semantic specificity [41] and the limited processing of the modeled information [13]. One of the most widely used EA modeling tools is Archi[1].

### 2.1 Enterprise Architecture Analysis

EA analysis concerns using EA models to analyze selected properties to provide decision support. Barbosa et al. [4] defined a taxonomy to classify EA research according to their analysis concerns, analysis techniques, and modeling languages employed to ease value extraction from EA models. A comprehensive survey of research on EA analysis techniques is presented by Buckl et al. [10]. The authors indicate a lack of automated analysis techniques that also scale well. A more recent survey yielded that *"Modern analysis approaches should combine interactive visualizations with automated analysis techniques"* [33]. The study by Santana et al. [47] reveals the need to develop proper tooling for EA analysis. Närman et al. [37] present a framework based on the ArchiMate metamodel for assessing four properties: application usage, system availability, service response time, and data accuracy. Florez et al. [19] present a catalog of automated analysis methods for enterprise models in a standardized modeling language and further implement the methods in a modeling tool. Domain ontologies have been applied for the representation, domain-knowledge enrichment, and analysis of EA models [14].

### 2.2 Graph-based Analysis of EA Models

Aside from the previously presented approaches that base the analysis on a specific EA framework or modeling language, we focus on the following approaches that utilize graph-based representation analysis of EAs. With increasing model size and complexity, ArchiMate models can get difficult to comprehend by humans. Graph visualizations can be compelling as they further abstract the different ArchiMate elements to the two basic concepts, i.e., *nodes* and *edges*. Graph visualizations can be easily customized. Furthermore, storing a graph in a graph database enables the efficient execution of complex queries over large graphs.

[1] https://www.archimatetool.com/, last accessed: 15.08.2022

Transforming EA models into graphs [4] or Linked Data [42, 31] to enable semantic analysis is not new. Such EA analysis focuses on quantitative graph theory, which measures (i.e., quantifies) structural aspects of graphs. Many quantitative graph measures exist like *PageRank* and *Betweeness* (cf. [17]). Caetano et al. [14] map the conceptual schemas of EA models to an (upper-level) ontology and present their further analysis through logical inference or graph analysis. Several works use graphs for maintaining and optimizing EAs. Giakoumakis et al. [23] replace existing services with new services while aiming not to disrupt the organization using multi-objective optimization on a graph representation of the EA model. Similarly, Franke et al. [22] use a binary integer programming model to optimize the relation between IT systems and processes. Prediction based on EA models, represented as graphs, has been proposed by MacCormack et al. [35] using Design Structure Matrices to analyze the coupling between EA components and Hacks and Lichter [26] using a probabilistic approach that considers different scenarios. Holschke et al. [29] perform failure impact analysis with Bayesian Belief Networks, and Buschle et al. [13] adapt ArchiMate by fault trees to analyze the availability of EA components. Plataniotis et al. [43] present decision design graphs to analyze, e.g., how the decisions taken on the business level affect decisions on the technology level.

### 2.3 Knowledge Graphs and Enterprise Architecture

Knowledge Graphs (KGs) have, since their popularization by Google in 2012, seen widespread adoption in academia and industry. They have been used to derive "world knowledge graphs" such as Google's KG, or DBpedia [2], but also "enterprise knowledge graphs" that represent more specific application domains [5]. In the case of this paper, the domain is EA itself.

KGs have been applied to represent different kinds of conceptual models like genomic datasets [7]. The Resource Description Framework (RDF) is the most common representation of KGs, and transformations from conceptual models to RDF have been proposed [52]. KG-based representation of models also enables reasoning methods for KGs, including logic-based and machine learning-based reasoning. Examples include supporting the analytic process [46] and more general reasoning contexts for conceptual modeling [36]. Bakhshadeh et al. [3] proposed the transformation of ArchiMate models into a Web Ontology Language (OWL) representation that enables consistency and completeness analysis of the EA models. However, OWL-based reasoning does not utilize the graph-based structural properties of EA models for analysis.

More recently, the first works proposed transforming EA models into KGs [50, 49] with initial experimental results toward using the KG for EA Smell detection [51]. This current paper extends this stream of research by first proposing a generic model-based Knowledge Graph construction process (see Section 3). While previous works concentrated on transforming the syntactic nature of EA models into equivalent KGs, in the proposed process, we emphasize how to augment knowledge of the KG from graph algorithms and EA Smells analysis by following an approach that comprises *knowledge creation*, *knowledge enrichment*,

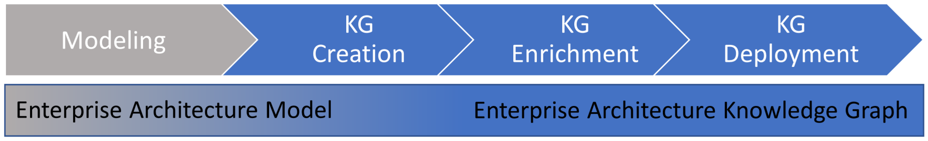



Fig. 1: Model-based Knowledge Graph Construction Process

and *knowledge deployment* phases. As a final extension, we implement tool support for our approach using an Archi plugin (see Section 4).

## 3 Model-based EA Knowledge Graph Construction

Various approaches exist for structuring the creation and the life cycle of KGs (cf. [6, 48, 40]). While no definitive procedure exists yet, many of the referred approaches focus on some form of *creation*, *evolution/enrichment*, and *deployment/use*. In this work, we introduce a **Model-based Knowledge Graph Construction Process** (see the upper part of Fig. 1), which is applied to the EA domain in order to construct an Enterprise Architecture Knowledge Graph (EAKG) (see the lower part of Fig. 1). The process structures the transition from EA modeling toward constructing a KG that enables representation, enrichment, and reasoning of EA knowledge. Initially, knowledge is extracted and created from the EA models in the **KG Creation** phase. Further knowledge is enriched, and additional inferences are made using that knowledge in the **KG Enrichment** phase. Finally, **KG Deployment** tailors the resulting KG to specific applications and facilitates different reasoning and representation approaches.

### 3.1 Knowledge Graph Creation

In the KG Creation stage, the EA model's relevant information is extracted to create the initial EAKG. Conceptual models follow a schema and provide metadata (e.g., naming and classification) for objects, relations, and properties most often specified by metamodels [9]. This information can facilitate the KG creation stage by mapping it to specific nodes, edges, and properties in the EAKG. The specificity of models and the expressiveness of the used modeling language here clearly plays a significant role in the quality and richness of the initial KG. The transformation of languages that already conform to a graph structure (such as ArchiMate) into a KG structure is straightforward, while for other languages, a deeper investigation of meaningful mappings is necessary.

In our work, we use ArchiMate models and map the ArchiMate metamodel to the KG metamodel (see Fig. 2). The knowledge graph metamodel is inspired from [16]. The ArchiMate metamodel consists of different kinds of elements that are structurally categorized. A relationship is divided into a *structural*, *dependency*, *dynamic*, or *others* category. An EA relationship is mapped to the *edge* of the KG metamodel. A relationship connector is a property of an EA relationship; therefore, it is mapped to the property of the KG metamodel. The concrete

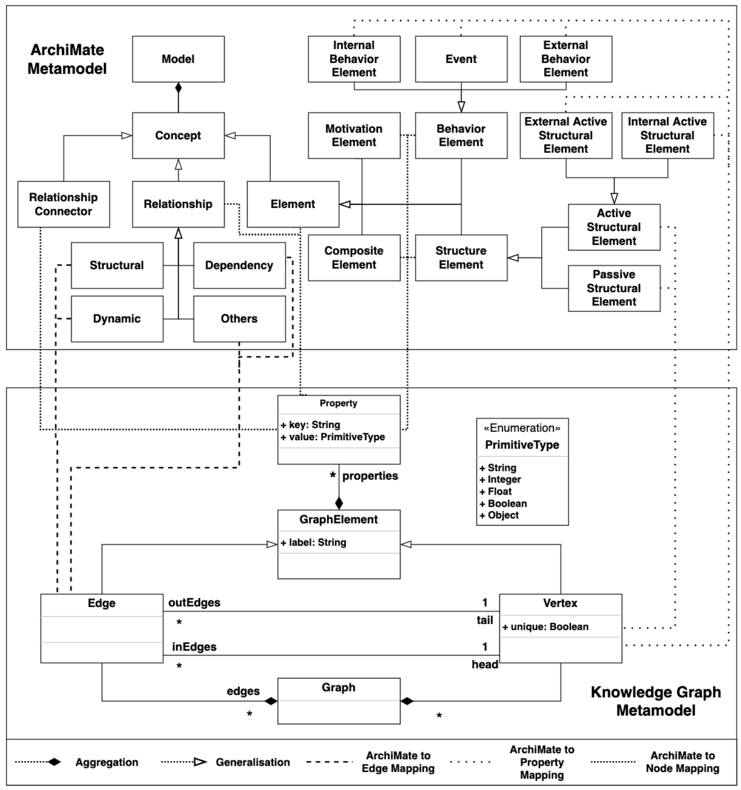



Fig. 2: ArchiMate behavioral and structural elements to KG metamodel mapping (KG metamodel inspired from [16])

behavioral and structural elements are mapped to the *node*. The abstract details of elements are mapped to the *property* of KG metamodel. For example, an *Active Structural Element* will be mapped to a node of KG metamodel, whereas it inherits the properties of a *Structure Element* and therefore, *Structure Element* is mapped to the property of KG metamodel. EAKG, using this mapping, thereby captures the semantics of EA model elements in the properties, and the structural elements are mapped to the nodes and edges of the KG.

Hoefferer [28] introduced the notion of *type semantics* and *inherent semantics* as two contributing aspects to derive the "full semantic description of model elements". Type semantics is defined by the metamodel concepts and their properties themselves. Therefore, this kind of semantics is invariant to all instantiations and applies transitively to them. Once modelers create a model by instantiating metamodel concepts, they define the inherent semantics. These kinds of semantics are contingent on the modeler and the modeled case. The ArchiMate generic framework comprises different layers for representing different enterprise viewpoints. Each layer comprises active, passive, behavioral, and motivational aspects. Each aspect is formed by different elements shown in the ArchiMate

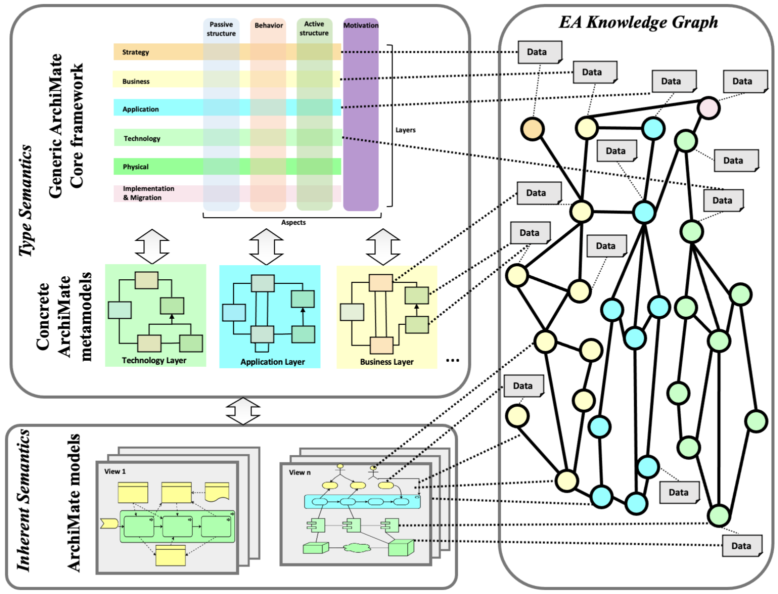



Fig. 3: EAKG Creation

metamodel in Fig. 2. Fig. 3 visualizes our approach of transforming ArchiMate models into a property graph-based EAKG using the mappings from Fig. 2. The EA to KG mapping thereby, incorporates the type and inherent semantics from EA models. From the generic ArchiMate framework, we can transform metadata like ArchiMate *layers* and *aspects* (shown in different colors in Fig. 3) into properties of nodes and edges of the EAKG. We can further derive type semantics from the concrete ArchiMate layer-specific metamodels like the *Application Layer* metamodel. We derive the inherent semantics from the concrete ArchiMate models, like the elements' names and connections.

### 3.2 Knowledge Graph Enrichment

The next step of the KG construction process is focused on knowledge enriching through general and domain-specific knowledge. Enriching such knowledge results in additional labels, properties, and edges in the EAKG. In this work, we enrich the initial EAKG by *Graph characteristics* (i.e., general graph-theoretical knowledge) and *EA Smells* (i.e., domain-specific EA knowledge). Hacks et al. introduce the concept of EA Smells [25] analogous to Code Smells [20] in the software engineering domain. EA Smells signify bad modeling practices and allow architects to discover possible flaws in their models. A catalog of EA Smells was published [45], and the website[2] serves as a knowledge base, currently listing 63 EA smells.

**Graph characteristics**. Graph characteristics describe quantitative aspects of the KG regarding structural characteristics (e.g., *centralities* and *communi-*

[2] https://swc-public.pages.rwth-aachen.de/smells/ea-smells/, accessed: 11.05.2022

*ties*). The initial graph expands with new properties for nodes, representing the score of specific graph algorithms. The score property can then be used for new KG representations, e.g., *node size* to highlight centralities or *color* to differentiate communities (exemplified in Section 5).

**EA Smells**. An approach for KG-based EA Smell Detection has been introduced in [51]. We adopt some of the proposed smell detection queries and use them during the knowledge enrichment step to enrich the EAKG with EA Smell-specific knowledge. Detected smells expand the EAKG structure with labels on affected nodes or by adding relationships when multiple objects are affected by a smell (e.g., introducing a new edge with the label 'duplication' between two nodes in case the *Duplication* smell detected duplicate nodes in the EAKG).

### 3.3 Knowledge Graph Deployment and Application

This step focuses on deploying the EAKG and allowing it to power various applications and use cases. The complexity of the preceding steps is entirely concealed to the user, i.e., to enterprise architects. The deployment takes the entire EAKG and provides functionality to explore, represent, and reason the EA knowledge efficiently. The architecture and implementation details are provided in the tool paper [24].

Still, the user can identify the provenance of each node, property, or edge or limit the scope of the EAKG to individual parts of the enterprise architecture. Thus, the primary focus of the deployment step is to make the EAKG easily accessible while providing features to work with the knowledge enrichment (e.g., only represent EA elements of a particular layer, only show parts of an EA that are affected by an EA Smell). We describe these (and more) features in great detail throughout the remainder of this paper, but in particular in Section 5.

## 4 An EAKG Archi Plugin

The result of the first iteration of the DSR life cycle is an initial prototype, employing the discussed concepts of Section 3. In particular, the prototype realizes automated reasoning and representation of EA knowledge based on the previously introduced *Model-based Knowledge Graph Construction Process* as a plugin for the Archi modeling toolkit. The plugin aims to make Knowledge Graph-based EA analysis available to enterprise architects, i.e., an audience that not necessarily has graph-theoretic knowledge.

Fig. 4 shows the integration of the plugin within the Archi application, containing both the *Graph View*, and the *Smells Report View*. The node colors in the EAKG are derived from the ArchiMate core framework based on the EA layer. The used EA model is a modified version of the ArchiSurance case study – for details, see Section 5.

**Knowledge Graph Visualization**. The main view in the center visualizes the EAKG. Nodes denote ArchiMate elements, while edges denote ArchiMate relationships. The colors of nodes derived from the ArchiMate core framework.

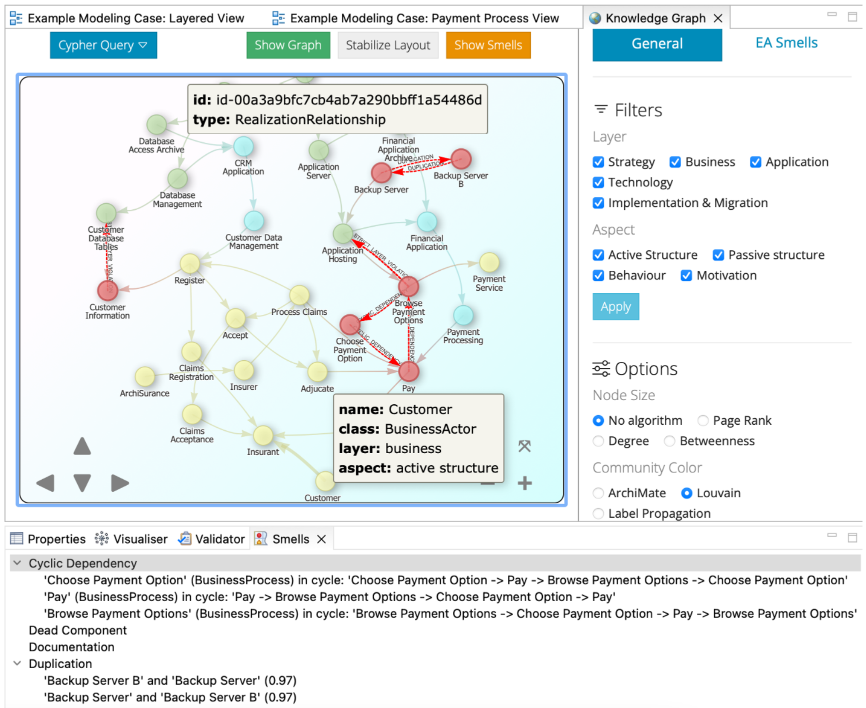



Fig. 4: EAKG plugin in Archi with Graph View and Smells Report View

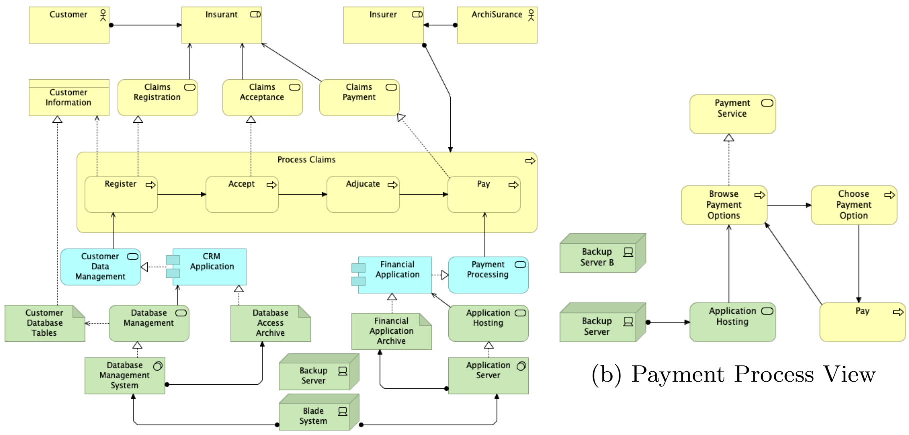
Further properties are exposed by hovering over nodes and edges, as exemplified by the *Customer* element in the figure and the relationship at the *Database Access Archive* and *CRM Application*. The toolbar at the top allows the execution of custom cypher queries on the EAKG.

**Graph characteristics visualisation**. The right-hand sidebar includes a filter and options menu. Here, enterprise architects can easily filter the displayed elements from specific layers or aspects of ArchiMate. The option menu on the bottom right offers configurations for the *Graph characteristics Knowledge Graph Enrichment* introduced in Section 3. Graph centrality measures are reflected via the node size, whereas community measures are reflected via node color.

**EA Smells Detection**. The Report view at the bottom lists all EA Smells and the affected elements in the model. The EA Smells view shows the affected elements highlighted in red and references to other elements of the smell represented as dashed, red edges. The EA Smells tab in the sidebar provides information about each EA Smell, including a visualization, a description, and a solution (for examples, see Section 5).

## 5 Case-based Evaluation

In order to evaluate the feasibility of constructing the EAKG (see Fig. 3) and using it as means of reasoning and representing EA knowledge, including EA




(a) Layered View

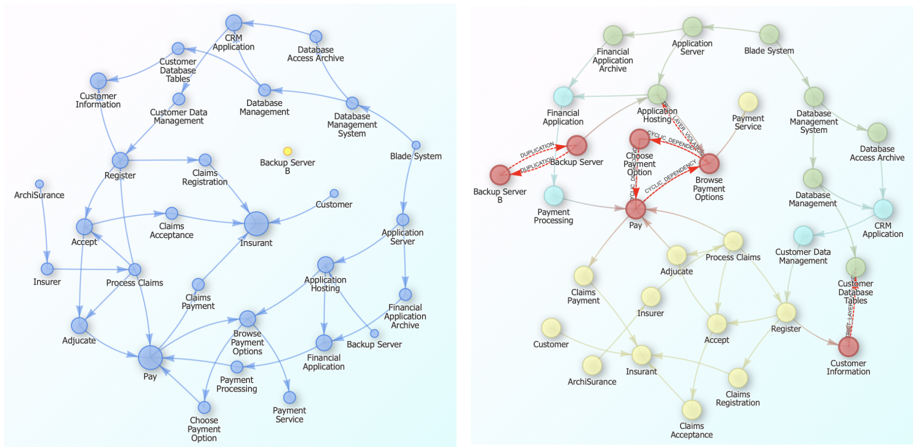
(b) Payment Process View

Fig. 5: Excerpts of the original EA model used during the case study (color of layers for model elements derived from the ArchiMate core framework)

Smells, we present a case-based evaluation of an ArchiMate model based on the popular ArchiSurance case study [30]. The model consists of multiple viewpoints, three of which are visualized in Fig. 5a: the *Layered Viewpoint*, which is reused from the ArchiSurance model and in the Fig. 5b *Payment Process Viewpoint*. Different colors in the models denote the various ArchiMate layers. The case study extends the ArchiSurance case by the requirements and realisation of online shopping and portfolio management of insurance products.

Fig. 4 already shows the resulting initial graph structure and how the ArchiMate elements and relationships are mapped to nodes and edges in the EAKG. The transformation maps the properties related to layers (e.g., *Strategy, Business*) and aspects (e.g., *Active Structure, Passive Structure*) in the original EA model to the properties of the nodes in the resulting EAKG. The relationship type (e.g., *realisation, assignment, association*) are stored in the properties of the relationships in the EAKG. Further information about the relationships is derived and stored as edge properties in EAKG; for example, realisation relationship is a structural relationship, and association relationship is a dependency relationship. Therefore, information about the relationship like *structural* or *dependency* is derived from the relationship type and stored as an edge property. In this way, the structural aspects are stored as nodes and edges and the semantic aspects of the EA elements (related to layers and aspects and further details about model elements) are stored as properties of nodes and edges in the EAKG. Next, we show the enrichment and analysis of the EAKG using Fig. 6.

*Graph-based Analysis.* After the KG construction, the EAKG can be further analyzed and enriched by applying graph algorithms. Fig. 6(a) visualises the resulting graph after executing the *Weakly Connected Components* graph algorithm. *Node Size* is set to *Degree* and the *Community Color* to *Weakly Connected Components*. The degree denotes the number of connections, and,



(a) Graph characteristics analysis view (b) Smell detection view

Fig. 6: KG-based EA Analysis Representations in EAKG

as can be seen, the size of nodes increases with the amount of incoming and outgoing edges. The Weakly Connected Components algorithm helps identify disconnected sub graphs by assigning each node that is part of the same sub graph with the same color. In our case, the graph consists of two disconnected subgraphs with different colors each. The degree of the node denotes the importance of each node. In our case, the Insurant can be seen as an important node because of its biggest size and provides insights to the modelers about checking the incoming and outgoing edges.

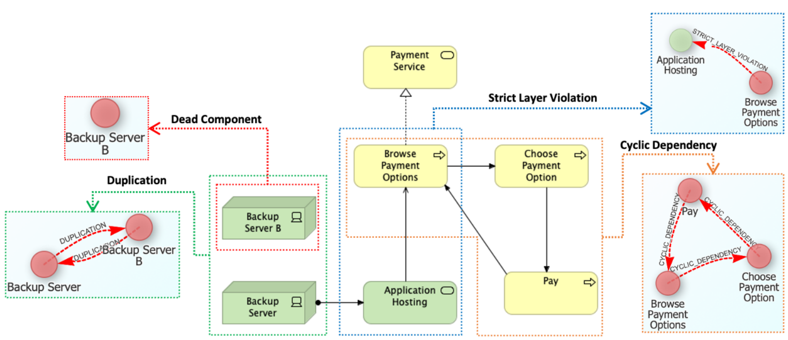



Fig. 7: EA Smells

*EA Smell Detection.* The EA model of our case was designed to include different EA Smells to showcase their representation in the enriched KG. We, e.g., added a *Long Documentation* text to the ArchiSurance element (in the Layered Viewpoint). Throughout the case we further added *Dead Component*, *Strict Layer Violation*, and *Cyclic Dependencies* smells. As can be seen in Fig. 6(b), nodes that are part of a detected EA Smell are highlighted in red, while edges to other nodes (that are also part of the detected EA Smell) are represented by a newly introduced dashed red edge with the name of the detected EA Smell as a label. The present smells in the viewpoints are displayed in Fig. 7, with individual representations mapping for each smell to the source model elements. Note that the EA Smells and the KG-based EA Smell detection have been introduced previously. We refer the interested reader to the dedicated literature [45, 51]. We contribute here a much richer representation of EA Smells that again uses a Knowledge Graph instead of a textual analysis proposed in previous research.

## 6 Conclusive Discussion

Our approach and the toolkit enable many possibilities for generalization (e.g., to other modeling languages) and extension (e.g., to incorporate further EA Smells or other knowledge). It is important to note that the metadata of an EA model captured by EAKG (e.g., metamodel level properties of the elements and relationships) are not fully utilized in the presented analysis; however, our EAKG transformation enables the possibility of applying KG-based techniques to analyze and process the EA models and further support the modelers. Ontologies can be integrated into the EAKG; therefore, foundational or domain ontologies can be linked to the EAKG for knowledge enrichment of the EA models and further apply reasoning. The conceptual schema of the EA models can be mapped to an ontology for knowledge enrichment and annotation for e.g., a health domain ontology can enrich the semantics of EA models of a hospital.

EAKG enables the KG-based AI applications for the semantic processing of EA models to support reasoning (e.g., inference-based, machine learning-based), integration (e.g., ontology mapping). EAKG provides the feasibility of applying machine learning techniques. The semantic relationships between model instance data and the models, along with the labels and the metadata (metamodel labels), can use NLP to predict links from a model element to an ontology element. Models can be mapped to common ontologies to further support interoperability. Graph Neural Networks (GNN) support applications such as node, edge, and even graph classification, link prediction between entities [53]. GNNs has been applied for UML model completion [18]. Similarly, GNNs with NLP techniques can be applied on EAKGs to support element recommendation or model autocompletion during modeling. GNNs can further transform an EAKG into a vector space with encoded specific domain information, enabling a domain-specific EA model semantic search.

In order to cope with the increasing complexity of maintaining and analyzing overarching enterprise architecture models, in this paper, we proposed an approach for model-based Enterprise Architecture Knowledge Graph (EAKG) construction and means to enrich type semantics, inherent semantics, general graph knowledge, and domain-specific enterprise architecture knowledge into the resulting EAKG. To evaluate our approach's feasibility and make it available to enterprise architects, we developed a first tool prototype, an EAKG plugin to the widely used Archi toolkit. Our approach enables full automation for the entire EAKG construction process and provides an efficient and intuitive GUI to explore and analyze the enterprise architecture knowledge graph.

The most innovative contribution we make with this paper is the not trivial enrichment of the EAKG from multiple sources and using the KG for EA analysis and representing EA knowledge using, e.g., the added nodes and relationships for EA Smells. Consequently, we propose to not only use KGs for automated analysis of overarching EA models, but also to improve human understandability by appropriate graph visualizations.

In this paper, we report on the results of a first Design Science Research cycle that aims at integrating all relevant sources into a fully-packed plugin archive. Future prototypes will emphasize a more lightweight plugin that interconnects the EA Smells catalog [45]. Instead of hard-coding the smell detection queries, such integration would enable us to always use the latest set of EA Smells together with their detection queries. Such a distributed system, of course, requires adequate infrastructure and latency, so we intended not to go along that path in developing the prototype. Moreover, instead of integrating a neo4j database, the plugin could easily connect to an existing neo4j instance.

A limitation of this research is the fact that we present a single case in this paper – previous works showed the scalability of the Graph-based EA Smell detection [51]. Further, as we aim to support enterprise architects with our approach, too, we need to engage in empirical evaluations to test the hypothesis on the perceived usefulness, ease of use, and intention to use EAKG in practice. Indeed, we are currently in discussions with a German-based international company on using and evaluating our approach.

As we know that showing larger models or graphs in a paper format might limit comprehensibility, we also created a demo video showcasing the EAKG Archi plugin in action. The video shows the core functionality and the case study example in detail and is accessible via: https://youtu.be/gcXiAWDJDes. The implementation of the EAKG plugin is open source[3] and we aim to list it on the Archi plugins page for researchers, teachers and EA practitioners[4].

## Acknowledgements

This work has been partially funded through the Erasmus+ KA220-HED project "Digital Platform Enterprise" (DEMO) with the project number: 2021-1-RO01-KA220-HED-000027576, the project "Enterprise Architecture Knowledge Graphs" funded by a Career Grant of TU Wien, and the Austrian Research Promotion Agency (FFG) via the Austrian Competence Center for Digital Production (CDP) under the contract number 854187.

[3] EAKG Github repository: https://github.com/borkdominik/archi-kganalysis-plugin
[4] Archi plugins: https://www.archimatetool.com/plugins/, accessed 02.05.2022

## References


1. Ahlemann, F., Stettiner, E., Messerschmidt, M., Legner, C.: Strategic enterprise architecture management: challenges, best practices, and future developments. Springer Science & Business Media (2012)
2. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.G.: Dbpedia: A nucleus for a web of open data. In: Aberer et al. (ed.) The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007. LNCS, vol. 4825, pp. 722–735. Springer (2007)
3. Bakhshadeh, M., Morais, A., Caetano, A., Borbinha, J.: Ontology transformation of enterprise architecture models. In: Doctoral Conference on Computing, Electrical and Industrial Systems. pp. 55–62. Springer (2014)
4. Barbosa, A., Santana, A., Hacks, S., Stein, N.v.: A taxonomy for enterprise architecture analysis research. In: 21st International Conference on Enterprise Information Systems. vol. 2, pp. 493–504. SciTePress (2019)
5. Bellomarini, L., Fakhoury, D., Gottlob, G., Sallinger, E.: Knowledge graphs and enterprise AI: the promise of an enabling technology. In: 35th IEEE International Conference on Data Engineering. pp. 26–37. IEEE (2019)
6. Bellomarini, L., Sallinger, E., Vahdati, S.: Knowledge graphs: The layered perspective. In: Knowledge Graphs and Big Data Processing, pp. 20–34. Springer (2020)
7. Bernasconi, A., Canakoglu, A., Ceri, S.: From a conceptual model to a knowledge graph for genomic datasets. In: Laender et al. (ed.) 38th International Conference on Conceptual Modeling. pp. 352–360. LNCS, Springer (2019)
8. Bork, D., Gerber, A., Miron, E.T., van Deventer, P., Van der Merwe, A., Karagiannis, D., Eybers, S., Sumereder, A.: Requirements engineering for model-based enterprise architecture management with archimate. In: Workshop on Enterprise and Organizational Modeling and Simulation. pp. 16–30. Springer (2018)
9. Bork, D., Karagiannis, D., Pittl, B.: A survey of modeling language specification techniques. Inf. Syst. **87** (2020). https://doi.org/10.1016/j.is.2019.101425
10. Buckl, S., Matthes, F., Schweda, C.M.: Classifying enterprise architecture analysis approaches. In: IFIP-International Workshop on Enterprise Interoperability. pp. 66–79. Springer (2009)
11. Burgueño, L., Kessentini, M., Wimmer, M., Zschaler, S.: 3rd workshop on artificial intelligence and model-driven engineering. In: Int. Conference on Model Driven Engineering Languages and Systems Companion. pp. 148–149 (2021)
12. Buschle, M., Holm, H., Sommestad, T., Ekstedt, M., Shahzad, K.: A tool for automatic enterprise architecture modeling. In: International Conference on Advanced Information Systems Engineering. pp. 1–15. Springer (2011)
13. Buschle, M., Johnson, P., Shahzad, K.: The enterprise architecture analysis tool - support for the predictive, probabilistic architecture modeling framework pp. 3350–3364 (2013)
14. Caetano, A., Antunes, G., Pombinho, J., Bakhshandeh, M., Granjo, J., Borbinha, J., Da Silva, M.M.: Representation and analysis of enterprise models with semantic techniques: an application to archimate, e3value and business model canvas. Knowledge and Information Systems **50**(1), 315–346 (2017)
15. Chen, X., Jia, S., Xiang, Y.: A review: Knowledge reasoning over knowledge graph. Expert Systems with Applications **141**, 112948 (2020)
16. Daniel, G., Sunyé, G., Cabot, J.: Umltographdb: mapping conceptual schemas to graph databases. In: International Conference on Conceptual Modeling. pp. 430–444. Springer (2016)

17. Dehmer, M., Emmert-Streib, F., Shi, Y.: Quantitative graph theory: A new branch of graph theory and network science. Information Sciences **418-419**, 575 – 580 (2017)
18. Di Rocco, J., Di Sipio, C., Di Ruscio, D., Nguyen, P.T.: A gnn-based recommender system to assist the specification of metamodels and models. In: Int. Conference on Model Driven Engineering Languages and Systems (MODELS). pp. 70–81 (2021)
19. Florez, H., Sánchez, M., Villalobos, J.: A catalog of automated analysis methods for enterprise models. SpringerPlus **5**(1), 1–24 (2016)
20. Fowler, M.: Refactoring: improving the design of existing code. Addison-Wesley Professional (2018)
21. Frank, U., Strecker, S., Fettke, P., vom Brocke, J., Becker, J., Sinz, E.J.: The research field "modeling business information systems" - current challenges and elements of a future research agenda. Bus. Inf. Syst. Eng. **6**(1), 39–43 (2014)
22. Franke, U., Holschke, O., Buschle, M., Narman, P., Rake-Revelant, J.: It consolidation: an optimization approach. In: Int. Enterprise Distributed Object Computing Conference Workshops. pp. 21–26 (2010)
23. Giakoumakis, V., Krob, D., Liberti, L., Roda, F.: Technological architecture evolutions of information systems: Trade-off and optimization. Concurrent Engineering **20**(2), 127–147 (2012)
24. Glaser, P.L., Ali, S.J., Sallinger, E., Bork, D.: Exploring enterprise architecture knowledge graphs in archi: The eakg toolkit (2022), under review
25. Hacks, S., Höfert, H., Salentin, J., Yeong, Y.C., Lichter, H.: Towards the definition of enterprise architecture debts. In: 2019 IEEE 23rd International Enterprise Distributed Object Computing Workshop (EDOCW). pp. 9–16. IEEE (2019)
26. Hacks, S., Lichter, H.: A probabilistic enterprise architecture model evolution. In: Int. Enterprise Distributed Object Computing Conference. pp. 51–57 (2018)
27. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design science in information systems research. MIS Q. **28**(1), 75–105 (2004)
28. Höfferer, P.: Achieving business process model interoperability using metamodels and ontologies. In: Österle, H., Schelp, J., Winter, R. (eds.) European Conference on Information Systems, ECIS 2007. pp. 1620–1631 (2007)
29. Holschke, O., Närman, P., Flores, W.R., Eriksson, E., Schönherr, M.: Using enterprise architecture models and bayesian belief networks for failure impact analysis. In: Int. Conference on Service-Oriented Computing. pp. 339–350 (2008)
30. Jonkers, H., Band, I., Quartel, D.: The archisurance case study. The Open Group pp. 1–32 (2012)
31. Karagiannis, D., Buchmann, R.A.: Linked open models: Extending linked open data with conceptual model information. Inf. Syst. **56**, 174–197 (2016)
32. Lankhorst, M.M.: Enterprise Architecture at Work - Modelling, Communication and Analysis, 2nd Edition. The Enterprise Engineering Series, Springer (2009)
33. Lantow, B., Jugel, D., Wißotzki, M., Lehmann, B., Zimmermann, O., Sandkuhl, K.: Towards a classification framework for approaches to enterprise architecture analysis. In: Proceedings PoEM 2016. pp. 335–343 (11 2016)
34. Maass, W., Storey, V.C.: Pairing conceptual modeling with machine learning. Data Knowl. Eng. **134**, 101909 (2021)
35. Maccormack, A.D., Lagerstrom, R., Baldwin, C.Y.: A methodology for operationalizing enterprise architecture and evaluating enterprise it flexibility. Harvard Business School working paper series# 15-060 (2015)
36. Medvedev, D., Shani, U., Dori, D.: Gaining insights into conceptual models: A graph-theoretic querying approach. Applied Sciences **11**(2), 765 (2021)

37. Naranjo, D., Sánchez, M., Villalobos, J.: Primrose: a graph-based approach for enterprise architecture analysis. In: International Conference on Enterprise Information Systems. pp. 434–452. Springer (2014)
38. Närman, P., Buschle, M., Ekstedt, M.: An enterprise architecture framework for multi-attribute information systems analysis. Software & Systems Modeling **13**(3), 1085–1116 (2014)
39. OMG: ArchiMate® 3.1 Specification. The Open Group (2019), http://pubs.opengroup.org/architecture/archimate3-doc/
40. Pan, J.Z., Vetere, G., Gómez-Pérez, J.M., Wu, H. (eds.): Exploiting Linked Data and Knowledge Graphs in Large Organisations. Springer (2017)
41. Pittl, B., Bork, D.: Modeling digital enterprise ecosystems with archimate: a mobility provision case study. In: Int. Conference on Serviceology. pp. 178–189 (2017)
42. Pittl, B., Fill, H.: Transforming enterprise models to linked data via semantic annotations. In: Schaefer, I., Karagiannis, D., Vogelsang, A., Méndez, D., Seidl, C. (eds.) Modellierung 2018. pp. 55–70. LNI, Gesellschaft für Informatik (2018)
43. Plataniotis, G., de Kinderen, S., Proper, H.A.: Relating decisions in enterprise architecture using decision design graphs. In: 2013 17th IEEE International Enterprise Distributed Object Computing Conference. pp. 139–146. IEEE (2013)
44. Reimer, U., Bork, D., Fettke, P., Tropmann-Frick, M.: Preface of the first workshop models in AI. In: Companion Proceedings of Modellierung 2020 Short, Workshop and Tools & Demo Papers. pp. 128–129. CEUR Workshop Proceedings (2020)
45. Salentin, J., Hacks, S.: Towards a catalog of enterprise architecture smells. In: Gronau, N., Heine, M., Krasnova, H., Poustcchi, K. (eds.) Internationalen Tagung Wirtschaftsinformatik, Community Tracks. pp. 276–290. GITO Verlag (2020)
46. Santana, A., Fischbach, K., de Moura, H.P.: Enterprise architecture analysis and network thinking: A literature review. In: Bui, T.X., Jr., R.H.S. (eds.) 49th Hawaii International Conference on System Sciences. pp. 4566–4575. IEEE (2016)
47. Santana, A., Simon, D., Fischbach, K., de Moura, H.: Combining network measures and expert knowledge to analyze enterprise architecture at the component level. In: 2016 IEEE EDOC Conference. pp. 1–10. IEEE (2016)
48. Simsek, U., Angele, K., Kärle, E., Opdenplatz, J., Sommer, D., Umbrich, J., Fensel, D.: Knowledge graph lifecycle: Building and maintaining knowledge graphs (2021)
49. Smajevic, M., Bork, D.: From conceptual models to knowledge graphs: a generic model transformation platform. In: Int. Conference on Model Driven Engineering Languages and Systems Companion. pp. 610–614 (2021)
50. Smajevic, M., Bork, D.: Towards graph-based analysis of enterprise architecture models. In: International Conference on Conceptual Modeling. pp. 199–209 (2021)
51. Smajevic, M., Hacks, S., Bork, D.: Using knowledge graphs to detect enterprise architecture smells. In: Practice of Enterprise Modeling. pp. 48–63. Springer (2021)
52. Tong, Q., Zhang, F., Cheng, J.: Construction of rdf (s) from uml class diagrams. Journal of computing and information technology **22**(4), 237–250 (2014)
53. Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., Sun, M.: Graph neural networks: A review of methods and applications. AI Open **1**, 57–81 (2020)
54. Zou, X.: A survey on application of knowledge graph. In: Journal of Physics: Conference Series. vol. 1487, p. 012016. IOP Publishing (2020)