Architecture Knowledge Graphs - a Next Step in Architecture Knowledge Management

Remco C. de Boer
ArchiXL & Vrije Universiteit Amsterdam
the Netherlands
rdeboer@archixl.nl
r.c.de.boer@vu.nl

ABSTRACT

This paper stems from a reflection on the state of the practice of the use of architecture knowledge especially within the Dutch public and semi-public sector. We have observed that publication of architecture knowledge in semantic wikis has helped organize the semi-structured nature of that knowledge as combinations of text and model elements, and has enabled sharing and connecting knowledge that typically lives within individual model repositories. Increasingly, however, we encounter the desire and need to make architectural knowledge break out of their isolated repositories. From several examples that illustrate this desire, this position paper argues that a next step in architecture knowledge management entails the use of linked data principles and techniques to construct an 'Architecture Knowledge Graph'.

CCS CONCEPTS

• **Information systems** → **Data structures**; *Enterprise information systems*; • **General and reference** → *Design*; • **Social and professional topics** → *Socio-technical systems*; • **Software and its engineering** → *Software design engineering*.

KEYWORDS

Architecture knowledge, Linked data, Knowledge graph

ACM Reference Format:
Remco C. de Boer. 2024. Architecture Knowledge Graphs - a Next Step in Architecture Knowledge Management. In *2024 International Workshop New Trends in Software Architecture (SATrends '24), April 14, 2024, Lisbon, Portugal.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3643657.3643914

1 INTRODUCTION

Architecture knowledge management has been a topic of research and practice for approximately two decades. Typical topics that have been addressed are the consideration of architecture as design decisions [10], the architecting process including architectural decision making [21] and biases, and codification of architectural knowledge for traceability, to evade knowledge evaporation and to promote reuse [1]. Over the years, many tools and techniques have been proposed, some of which have remained research prototypes and others that have had impact on the practice of architectural knowledge management.

This paper stems from a reflection on the state of the practice of the use of architecture knowledge especially within the Dutch public and semi-public sector. One tool that has gained particular traction there is the use of semantic wikis [3, 4]. One particular application is the publication of sector specific reference architecture knowledge with the intention that this information is reused by organizations within that sector. Such reference architectures address government services [9], municipalities [22], provinces[16], water control authorities [7], education [14, 17–19], and many more [8].

From experience within these sectors, we have observed that publication of architecture knowledge in semantic wikis has helped organize the semi-structured nature of that knowledge as combinations of text and model elements, and has enabled sharing and connecting knowledge that typically lives within individual model repositories. Increasingly, however, we encounter the desire and need to make architectural knowledge break out of their isolated repositories. Particularly for reference architectures within the Dutch public sector, recent research [15] shows the problems architects encounter and their drivers for improved coherence, but the desire for more explicit coherence and connection is not limited to reference architectures alone. This paper presents several examples of desired architecture knowledge connections encountered in practice. From these examples, this position paper argues that a next step in architecture knowledge management entails the use of linked data principles and techniques to construct an 'Architecture Knowledge Graph'.

This paper is organized as follows. Section 2 introduces the linked data paradigm and the concept of a knowledge graph. Section 3 provides examples encountered in practice that require architecture knowledge connections beyond a single architecture, illustrating the potential of an architecture knowledge graph. Section 4 discusses the gap between the current state of the practice and the realization of an architecture knowledge graph, outlining an agenda of possible future work.

2 LINKED DATA AND KNOWLEDGE GRAPHS

Linked data is a standards-based approach to structure, publish and connect data "so that a person or machine can explore the web of data. With linked data, when you have some of it, you can find other, related, data." [12] With linked data standards such as RDF [23] and SPARQL [6], data elements from various data sources can be technically and semantically connected. Linked data is built on four basic principles (or 'rules') coined by Tim Berners Lee [12]:

SATrends '24, April 14, 2024, Lisbon, Portugal

ACM ISBN 979-8-4007-0560-1/24/04
https://doi.org/10.1145/3643657.3643914

(1) Use URIs as names for things
(2) Use HTTP URIs so that people can look up those names.
(3) When someone looks up a URI, provide useful information, using the standards (RDF, SPARQL)
(4) Include links to other URIs, so they can discover more things.

2.1 RDF

The Resource Description Framework (RDF) [23] is a core standard for realization of the linked data principles. RDF provides a graph-based data model in which a graph consists of a set of *triples* that each consist of a *subject*, a *predicate*, and an *object*. Subjects and objects are nodes in the graph, and the predicates form relations between the nodes. Subjects denote entities in the universe of discourse, and are represented by an IRI [1] [2]. Like subjects, objects in an RDF triple can denote an entity from the universe of discourse - in which case they are represented by an IRI - or they can denote a 'literal'. The predicate in a triple denotes a property that expresses the relationship between the subject and object. A predicate takes the form of an IRI, for which the semantics are defined in an RDF Vocabulary.

2.2 SPARQL

SPARQL is a query language for RDF. The same triple pattern that is used in RDF, is used in SPARQL queries. In a SPARQL query, however, parts of the triple may be a variable.

SPARQL queries can be executed against a SPARQL endpoint which provides an interface to an RDF database or 'triple store', and can be federated over remote (distributed) SPARQL endpoints.

2.3 Knowledge graphs

Based on linked data principles and techniques, one can construct a so-called 'knowledge graph' [13] that spans multiple data sources and technically and semantically connects the different data entities. In this graph, individual data entities are the nodes, while the edges represent the semantic relation between those entities. The knowledge graph constituents - i.e., the data entities and relations between them - may live in multiple distributed data sources.

A graph-based representation of knowledge is in itself a logical fit for architecture models, which usually have a graph-like structure themselves e.g. as components-and-connectors. But a knowledge graph based approach in which data entities (including but not limited to architecture model elements) from different data sources can be semantically and technically connected may significantly increase the value of architecture knowledge.

3 EXAMPLES OF ARCHITECTURE KNOWLEDGE CONNECTIONS

To illustrate the potential of an architecture knowledge graph, this section shows some situations from practice in which connections beyond a single architecture can be seen.

3.1 Connections with enterprise and domain vocabularies

Elements in an architecture model represent objects that may be defined in an enterprise or domain vocabulary. These are controlled vocabularies or thesauri that are curated to provide accepted, and sometimes normative, definitions within a particular context. As an example, *Onderwijsbegrippen.nl* contains a vocabulary of terms and definitions for the Dutch education sector. These definitions are used in, for instance, reference models in the ROSA reference architecture for the education sector [18].

While many such vocabularies already conform to linked data principles and are for instance published in SKOS [2], reusing terms and definitions within an architecture model typically involves manually copying those to elements in the modelling environment.

3.2 Cross-architecture model connections

Cross-architecture connections may occur horizontally, between related architectures that use the same model kind, or vertically, between models within the same scope that use different model kinds.

Examples of **horizontal cross-architecture connections** can be found between the different reference architectures in the Dutch (semi-)public sector. While each of those reference architectures targets a specific sector - with its own particular rules, laws, and regulations that drive and shape it - these sectors are not strictly isolated and may exhibit similarities and even influence each other. Within the education sector, for instance, an advisory group[3] specifically addresses the need for further synchronization and harmonization between the different sector specific reference architectures for primary/secondary, vocational, and higher education. An analysis from this advisory group has shown how each of these reference architectures addresses learning resources, and how different concepts from the individual reference architectures are related [11]. Expressing those links *between* reference architectures in such a way that elements from different architecture models are traceable to each other is challenging, since these elements are confined to their own modelling and publication environments.

As for **vertical cross-architecture connections**, these typically occur when different - but related - architecture models express different aspects to address different goals or stakeholder concerns and have to do with traceability and model correspondence. An example thereof can be seen in the AMIGO approach[4] that is used within the education sector to design specifications for information exchange. This approach starts from a scope investigation together with business stakeholders, for which a conceptual model is expressed in ArchiMate [5]. This model shows at a high level which systems are involved, and which information flows between those system in the context of the business processes that are in scope of the collaboration. In a next iteration, the information flows are refined in finer grained interaction models that are expressed as UML sequence diagrams, and the information objects themselves are further refined in UML class diagrams. These are then input for

[1] IRIs are a generalization of URIs.
[2] Technically, subjects (and objects) can also be denoted by a construct called a 'blank node'. For the purpose of this discussion, the use of blank nodes is irrelevant.

[3] Adviesgroep Samenhang Onderwijsarchitecturen, https://www.edustandaard.nl/standaard_werkgroepen/adviesgroep-samenhang-onderwijsarchitecturen/
[4] https://www.edustandaard.nl/amigo/

a final iteration in which - using a particular UML profile - a UML class diagram is constructed that connects those classes to stereotyped classes that represent paths, parameters and REST actions. This model is subsequently used for model driven generation of an OAS specification that can be used by system implementors.

3.3 Reuse and refinement of reference architecture knowledge

The goal of reference architecture knowledge is to be reused and applied in concrete architectures or other reference architectures. Within the NORA family of reference architectures [8], architecture principles from NORA should be adhered to by 'daughter' architectures, and the 'inheritance' of those principles can manifest in different ways: a daughter reference architecture may cite a principle from NORA, make a verbatim copy of the principle to connect it - within their own environment - to elements from that daughter, or make an annotation to explain why that principle does not apply. This soon becomes problematic in the light of evolution of the architectures, since these relations between NORA and the inheriting architectures are difficult to trace. A similar problem occurs when sector-specific reference architectures are used as the basis for organization specific (enterprise) architectures, as there is no easy way to assess the effects of changes in the reference architecture on the organization specific architecture.

3.4 Operationalization of architecture knowledge

Another area in which connections beyond the architecture are required is in the operationalization of architecture knowledge. An example can be found in the Software Catalog application that originated in the municipality sector, and has since been adopted by several other sectors. The goal of the Software Catalog is to (a) provide an overview of software products that can be procured by individual organizations (e.g. municipalities, schools, water control authorities), (b) to provide a mapping of the capabilities of those software products according to their suppliers onto functionalities that have been specified in the sector reference architecture, and (c) to provide a means for organizations within the sector to register the specific functionalities they do or don't use for particular software products they have procured. The latter is meant as a way to share knowledge about actual software product usage within the sector, but also serves as a basis for individual organizations to quickly construct an overview of their 'application landscape' as a view on the underlying reference architecture model. As with the reuse and refinement of architecture knowledge outlined above, evolution of the reference architecture can lead to synchronization issues between these environments.

Another example of operationalization is the connection of operational data (monitoring data, application usage statistics) with architecture models. This transforms an architecture model into a dashboard that provides a 'live' view of what happens in the organization, and that can infer and visualize the impact of certain events [20], but also requires that a connection can be made between architecture knowledge - especially model elements -and other data sources.

4 TOWARDS AN ARCHITECTURE KNOWLEDGE GRAPH

In all the examples from Section 3 above, there is a need to connect elements from one architecture to another, or - in the case of operationalization - to link elements from another data source to elements from the architecture. The knowledge graph paradigm has serious potential to address those needs. In order to realize that potential, several non-trivial challenges need to be addressed. It is the author's intention to further investigate and address these challenges in future research.

Fundamentally, the ecosystem of architecture tools should become linked data-aware:

- Architecture tools should be capable of connecting to linked data (SKOS) vocabularies to traceably reuse concepts and definitions;
- Semantics of ADLs and model kinds need to be expressed as RDF Vocabularies;
- Modelling environments should be able to publish or export to RDF, and/or there should be transformations and mappings from native formats to standardized RDF vocabularies;
- Publication environments should provide SPARQL endpoints for model elements and other semantically enriched architecture knowledge (including design decisions, architecture principles, et cetera). They should also provide integration capabilities (i.e. federated queries) with - SPARQL endpoints of - other publication environments.

Upon this foundation, further knowledge management applications can be envisioned, such as

- Traceability services that derive relations between different models or architectures based on explicit cross-architecture connections;
- Evolutionary guidance and consistency checking by means of cross-architectural impact analysis of changes in related architectures;
- Compliance dashboards with respect to 'comply or explain'-adherence to architecture principles and design decisions;
- Alignment between architecture knowledge and other data sources, with further operationalization of architecture knowledge as a reusable and linkable asset for other environments.

5 AN EXAMPLE USE CASE

The *Onderwijsbegrippen.nl* vocabulary contains definitions for concepts used in the education sector, such as 'education participant', 'education provider', et cetera[5]. These concepts and definitions are reused in the educational reference architectures, where domain specific synonyms (e.g. 'student' or 'pupil' for 'education participant' in higher respectively primary education) may be applied. An education participant may have multiple occurrences in the reference architecture, e.g. as a 'Business Role' (cf. [5]) where the education participant is involved in a process, or as a 'Business Object' where information about the participant is processed or exchanged. Currently, architects have no other option then to copy

[5]The actual terms in this vocabulary are in Dutch

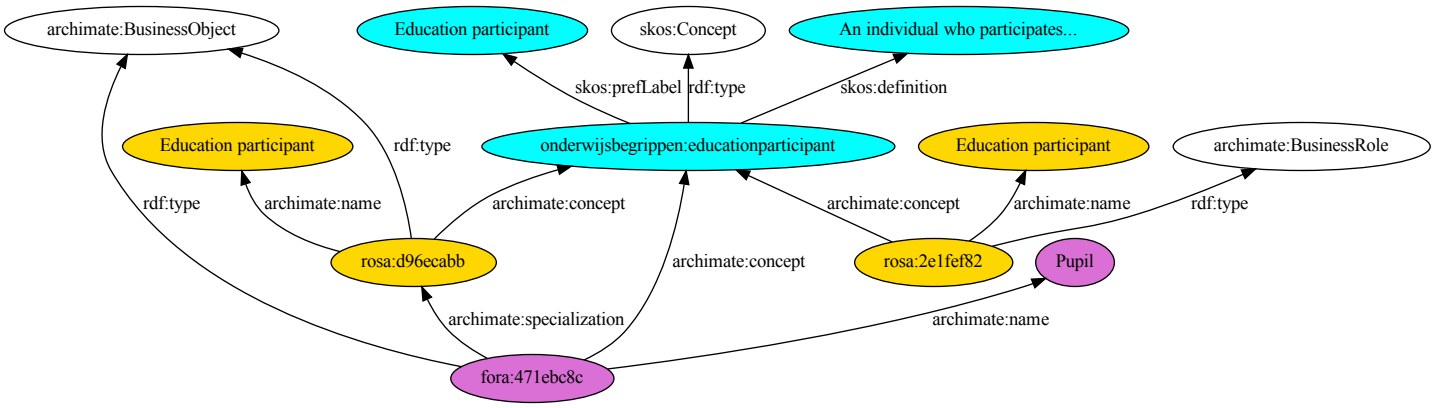

Figure 1: Example of an Architecture Knowledge Graph.

the definition from the vocabulary into the elements of their architecture model, and to manually synchronize changes in the vocabulary to the architecture model. A very first step to improve this situation would be to extend the architecture modelling tool (e.g. via a plugin) with a capability to connect elements from the architecture to the SKOS vocabulary and to automatically synchronize definition updates from the vocabulary to those elements. Likewise, it would be possible to connect model elements to elements from other architectures published as linked data. Eventually, with the right export and publication capabilities, this would lead to an architecture knowledge graph such as the one in Fig. 1. This graph contains connections between different repositories (blue: the Onderwijsbegrippen vocabulary, yellow: the ROSA reference architecture, purple: the FORA reference architecture), and traces e.g. the use of 'Pupil' in FORA to 'Education participant' in ROSA (as an example of a horizontal cross-architecture connection) and the definition of 'Education participant' in the domain vocabulary.

6 CONCLUSION

In conclusion, the construction of architecture knowledge graphs based on linked data principles is not trivial given that current architecture tools are typically not linked data-aware. Nevertheless, the knowledge graph paradigm has great potential to address relevant architecture knowledge management challenges we encounter in practice. Despite the required effort, architecture knowledge graphs should be a next step in architecture knowledge management.

REFERENCES

[1] Paris Avgeriou, Philippe Kruchten, Patricia Lago, Paul Grisham, and Dewayne Perry. 2007. Sharing and Reusing Architectural Knowledge–Architecture, Rationale, and Design Intent. In *29th International Conference on Software Engineering (ICSE'07 Companion)*. 109–110. https://doi.org/10.1109/ICSECOMPANION.2007.65

[2] Sean Bechhofer and Alistair Miles. 2009. *SKOS Simple Knowledge Organization System Reference*. W3C Recommendation. W3C. https://www.w3.org/TR/2009/REC-skos-reference-20090818/.

[3] Remco C. de Boer. 2017. ArchiMedes - Publication and Integration of Architectural Knowledge. In *2017 IEEE International Conference on Software Architecture Workshops (ICSAW)*. 268–271. https://doi.org/10.1109/ICSAW.2017.22

[4] Remco C. de Boer and Hans van Vliet. 2011. Experiences with Semantic Wikis for Architectural Knowledge Management. In *2011 Ninth Working IEEE/IFIP Conference on Software Architecture*. 32–41. https://doi.org/10.1109/WICSA.2011.14

[5] The Open Group. 2022. ArchiMate 3.2 Specification.

[6] Steven Harris and Andy Seaborne. 2013. *SPARQL 1.1 Query Language*. W3C Recommendation. W3C. https://www.w3.org/TR/2013/REC-sparql11-query-20130321/.

[7] Het Waterschapshuis. [n. d.]. WiLMA (Waterschap Informatie & Logisch Model Architectuur) - reference architecture for water control authorities. https://www.wilmaonline.nl/.

[8] ICTU. [n. d.]. NORA Familie - overview of reference architectures within the Dutch (semi-)public sector. https://www.noraonline.nl/wiki/NORA_Familie.

[9] ICTU. [n. d.]. NORA (Nederlandse Overheid Referentie Architectuur) - the Dutch Government Reference Architecture. https://www.noraonline.nl/.

[10] Anton Jansen and Jan Bosch. 2005. Software Architecture as a Set of Architectural Design Decisions. In *5th Working IEEE/IFIP Conference on Software Architecture (WICSA'05)*. 109–120. https://doi.org/10.1109/WICSA.2005.61

[11] Bas Kruiswijk. 2023. *Adviesdocument: Opbrengst focussessie leermiddelen*. Technical Report. Edustandaard, Adviesgroep Samenhang Onderwijsarchitecturen. https://www.edustandaard.nl/app/uploads/2023/02/Adviesdocument-ter-afstemming-Opbrengst-focussessie-leermiddelen.pdf.

[12] Tim Berners Lee. 2009. Design Issues - Linked Data. https://www.w3.org/DesignIssues/LinkedData.html.

[13] Sean Martin, Ben Szekely, and Dean Allemang. 2021. *The Rise of the Knowledge Graph: Toward Modern Data Integration and the Data Fabric Architecture*. O'Reilly Media.

[14] MBO Digitaal. [n. d.]. MORA - reference architecture for vocational education. https://mora.mbodigitaal.nl/.

[15] Remco M. Overvelde. 2024. Towards a method for improving the coherence between the reference architectures within the Dutch public sector.

[16] Platform Provincie Architecten. [n. d.]. PETRA (Provinciale Enterprise Referentie Architectuur) - reference architecture for provinces. https://petra.wikixl.nl/.

[17] Stichting Kennisnet. [n. d.]. FORA (Funderend Onderwijs Referentie Architectuur) - reference architecture for primary and secondary education. https://fora.wikixl.nl/.

[18] Stichting Kennisnet. [n. d.]. ROSA - reference architecture for the education domain. https://rosa.wikixl.nl/.

[19] SURF. [n. d.]. HORA (Hoger Onderwijs Referentie Architectuur) - reference architecture for higher education. https://hora.surf.nl/.

[20] Joeri C. van Es. 2019. Make Impact Visible: An impact analysis framework for Enterprise Architecture models expressed in the ArchiMate language. https://doi.org/10.13140/RG.2.2.24192.20484

[21] Hans van Vliet and Antony Tang. 2016. Decision making in software architecture. *Journal of Systems and Software* 117 (July 2016), 638–644.

[22] VNG Realisatie. [n. d.]. GEMMA (Gemeentelijke Model Architectuur) - reference architecture for municipalities. https://www.gemmaonline.nl/.

[23] David Wood, Markus Lanthaler, and Richard Cyganiak. 2014. *RDF 1.1 Concepts and Abstract Syntax*. W3C Recommendation. W3C. https://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/.