Structuring Scientific Knowledge in Software Engineering Using the Open Research Knowledge Graph: A Use Case and Experience Report

Angelika Kaplan[1], Fatma Chebbi[1], Dominik Fuchß[1], Oliver Karras[2], Tobias Hey[1], Anne Koziolek[1], and Ralf Reussner[1]

Abstract: Until now, papers have been the central communication medium for new research outcomes and findings that enhance the current body of scientific knowledge. In software engineering (SE), those research artifacts are aligned to further replication artifacts, comprising data and software underlying the empirical findings in a paper with validity and evidence. The research community is aware of the need for infrastructures with services that support the Findable, Accessible, Interoperable, and Reusable (FAIR) principles that foster and improve research data management. This implies a more convenient research environment and an easier knowledge transfer to research and practice. In this paper, we aim to explore the use of the Open Research Knowledge Graph (ORKG) for semantic structuring of research artifacts using the concept of ORKG template specification, supporting the aforementioned vision. Therefore, we present a use case in one of the research sub-fields of SE, namely software architecture and design (SWA). We explore the capabilities of the ORKG for describing semantic structures in this research field and discuss design decisions for the template specification. In addition, we present our developed tool Visulite which aims to contribute to a multi-modal knowledge graph like ORKG. Based on our work, we derive hints and share our insights with the research community. Moreover, we provide an open-access repository to maintain and document the different evolution steps w.r.t. our template specification for traceability, comprehension, and benchmarking purposes. In future work, we aim to maintain our evolved data schemas in SWA and the corresponding ORKG templates for the research community, fostering collaborations and benchmarking them to similar approaches to build consensus in SE research w.r.t. semantic structuring of research artifacts.

Keywords: Research Data Management, Software Engineering, Research Knowledge Graphs, Open Research Knowledge Graph, Semantic Modeling

1 Introduction

Software engineering (SE) is concerned with the systematic design, maintenance, and operation of software systems (cf. IEEE Std 610.12-1990 [IE90]). Consequently, SE research

[1] Karlsruhe Institute of Technology (KIT), KASTEL – Institute of Information Security and Dependability, Am Fasanengarten 5, 76131 Karlsruhe, Germany,
angelika.kaplan@kit.edu, https://orcid.org/0009-0009-9101-5833;
fatma.chebbi@alumni.kit.edu, https://orcid.org/0009-0008-7177-5213;
dominik.fuchss@kit.edu, https://orcid.org/0000-0001-6410-6769;
hey@kit.edu, https://orcid.org/0000-0003-0381-1020;
koziolek@kit.edu, https://orcid.org/0000-0002-1593-3394;
ralf.reussner@kit.edu, https://orcid.org/0000-0002-9308-6290

[2] TIB - Leibniz Information Centre for Science and Technology, Welfengarten 1B, 30167 Hannover, Germany,
oliver.karras@tib.eu, https://orcid.org/0000-0001-5336-6899

doi:10.18420/se2025-ws-29

should mainly support and improve these activities. The effects of such constructive research are reflected either in much higher product quality or improved process characteristics. For instance, the automation of an existing process can lead to higher product quality, lower costs, faster product results, or even a reduction in the required expertise. However, research outcomes in this field require empirical evidence to achieve the breakthrough into practice, indicating the maturity of this research field, which is covered and investigated by evidence-based software engineering (EBSE) [DKJ05; KDJ04]. To achieve this, optimal conditions in the research environments are crucial to cope with the resulting research artifacts, both data- and paper-wise. As a result, the research community is aware of the necessity for corresponding infrastructure. Providing services aligned with the FAIR principles is mandatory, as they enhance and facilitate the management of research data. In this regard, research knowledge graphs like ORKG [Ja19] can support the organization of scientific knowledge for communication and (re-) use for researchers and practitioners as well. In this paper, we aim to present a semantic structuring development approach in one of the SE sub-research field, namely software architecture (SWA) and design, using the concept of ORKG templates. We discuss experiences in developing such templates to provide hints for other research fields. Moreover, by using version tracking of derived templates, we can easily benchmark them against similar approaches and provide profound documentation of the evolution of templates in our research field. Besides facing challenges in semantic structuring w.r.t. research knowledge in text form, we present our tool Visulite (visualization of literature) [CFK23a], which enables the creation of graphical presentations based on raw data in replication packages. Consequently, the contributions of this paper are:

C1 We present our initial data schema for SWA research [Ko22a] and further extensions to transform it into an ORKG template. In addition, we share insights w.r.t. design decisions in the transformation process (from data schema to ORKG template).

C2 We introduce our open source tool Visulite [CFK23a; CFK23b] w.r.t. current features to easily contribute to a multi-modal knowledge graph and dashboards.

C3 We provide our artifacts w.r.t. the construction and maintenance of our ORKG templates in SWA research (i.e., tracking of versions and variants) in our open-access repository [KA24].

2 Research Knowledge Graphs

A research knowledge graph (RKG) is a structured representation of knowledge, enabling a semantic description by linking existing metadata and content data of scientific research artifacts (publications, software, and datasets). They build upon semantic web technologies and serve as a tool in research domains to enhance the efficiency of knowledge management and collaboration among researchers [RSS23]. RKGs improve the identification, traceability, and clarity of scientific concepts, reducing redundancy and duplication. Their structured representation also facilitates machine actionability that supports new digital services [Ka23].

The Open Research Knowledge Graph (ORKG) is one prominent RKG and an integral part of the NFDI (National Research Data Infrastructure) in Germany, organizing scientific information. The ORKG provides an open, ready-to-use infrastructure for organizing and curating, and maintaining scientific information [Ja19]. For this purpose, the ORKG structures scientific information into ORKG contributions, which are semantic descriptions of the information. ORKG contributions can be structured using ORKG templates to ensure consistent and comparable descriptions. In particular, ORKG templates define data structures, formats, and constraints that the ORKG can validate during data entries to ensure quality, integrity, and interoperability. Research knowledge in ORKG is described following the RDF (Resource Description Framework) subject-predicate-object paradigm. The object and the subject can contain resources and classes. The predicate contains properties. Literals (atomic pieces of information) are stored as objects. Thus, users can add their own additional predicates (properties or attributes) [Ja19].

3 Semantic Structuring Use Case: Software Architecture Research

In the following sections, we first present the initial data schema, describe the mapping process to the specification of the corresponding ORKG template, and finally discuss the advantages and disadvantages of our design decisions in the transformation process w.r.t. maintainability (esp., in terms of extensibility), usability aspects as well as suitability for intended purposes (e.g., knowledge retrieval). Lastly, we introduce our tool Visulite.

3.1 A Data Schema for Software Architecture Research

As indicated in the previous sections, empirical research creates a solid basis and evidence for the validity of research outcomes, builds comparability of research works, and indicates the maturity of a research field. As investigated by Galster and Weyns [GW16], the research field of software architecture (SWA) has experienced a remarkable increase in empirical studies as well. However, consensus about empirical research (e.g., reproducibility, applicability, and description of findings and evidence) in this research field is still limited [GW23].

Consequently, we proposed a data schema [Ko22a] in a top-down approach, based on the considerations and insights of our previous work (cf. [KWH21]), to foster a common understanding of how to describe research in SWA. This schema was validated with a literature review with at least two reviewers per paper (i.e., 153 full technical papers from 2017 to 2021 published at the International and European Conference on Software Architecture, namely ICSA and ECSA) in a bottom-up approach to show the feasibility of the data schema and provide an overview of current state-of-practice in SWA research. To support evolution scenarios of the schema, we specified a maintenance process, as introduced in [KKR22], using a hybrid approach (i.e., combining top-down and bottom-up construction and validation techniques).

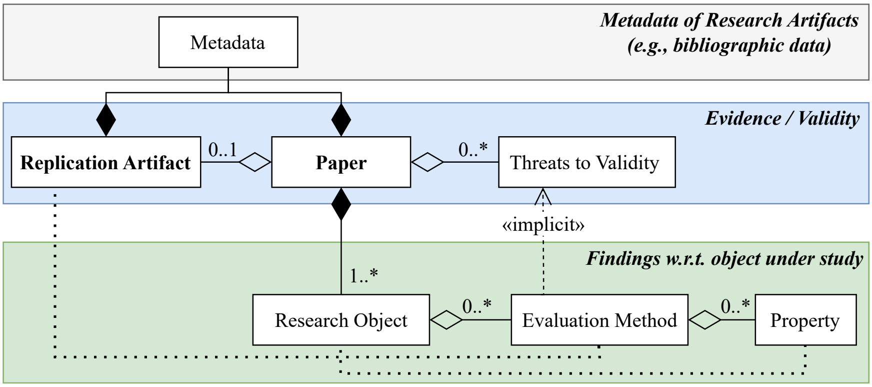

Fig. 1: Data schema and model for software architecture research, supporting semantic structuring (adapted and extended from [Ko22a])

The data schema for SWA research focuses on the description of research artifacts, namely research papers and corresponding replication artifacts. Both research artifacts can be further specified according to metadata (data about research artifact data) and content data. *Metadata* of papers refers to bibliographic information (e.g., research field, venue, title, or DOI as a unique identifier). Likewise, *Metadata* of replication artifacts refers to subdivided artifact units like software, tool support, or input data. Content data refers to the more generic description for research by characterizing findings (i.e., how researchers enhance the current body of knowledge) and their evidence. In this way, we assume to achieve generalizability to other research sub-fields in SE. Findings are characterized through a *Research Object* (i.e., the object under study), *Evaluation Method* (i.e., the method used to investigate the research object), and the *Property/ies* (i.e., investigated characteristics of a research object using an evaluation method). A paper has at least one research object. We derived 13 disjoint SWA research objects (e.g., *Architecture Analysis Method* or *Architecture Optimization Method*) [Ko22a]. As a baseline for the evaluation method, we refer to the ACM Empirical Standards [Ra20], enhancing this list of methods with domain-specific techniques like *Motivating Example* or *Technical Experiment* as identified in the aforementioned literature corpus. The final list of evaluation methods consists of 13 disjoint classes. For the property, the main authors chose the classification of ISO 25010 (version 2011) [IS11] in terms of *Quality in Use* (with 5 disjoint classes) and *Product Quality* (with 8 disjoint classes) as well as a classification of Taverniers et al. [TDV04] in terms of *Criteria for Analytical Methods* (with 9 disjoint classes). In this regard, further property categories (and classes within categories) may be added to the data schema when time evolves. Likewise, evidence (to the findings) is characterized via *Threats to Validity* (definitions of 6 disjoint classes adopted of Feldt and Magazinius [FM10]), that are implicitly specified by the design and context settings of the evaluation method(s) applied and via the provision of *replication* *artifacts*. In conclusion, Figure 1 depicts the main elements and relations of the data schema for describing SWA research.

3.2 An ORKG Template for Software Architecture Research Objects

The SWA data schema was directly mapped to the ORKG template concept in a linear transformation process. Therefore, we use the hierarchical structure in the data extraction process and the finite set of concrete instances for every main element in the data schema. Moreover, we have to consider the relations and cardinalities w.r.t. the description of the findings, which leads to a more complex data structure. The GitLab-wiki[3] of the replication package for the data schema study [Ko22a] provides an overview of the concrete data extraction format for the literature review.

In conclusion, Figure 2 depicts a short version of the ORKG template for SWA research based on the data schema of [Ko22a]. Colored boxes denote the correspondence of elements in the data schema (Figure 1) and the resulting ORKG template. The yellow boxes denote extensions of the data extraction format of the literature review, considering metadata aspects of replication artifacts that determine version 1.1 of our specified ORKG template. For a complete documentation, please refer to our aforementioned open-access repository.

3.3 Design Decisions in the Transformation Process

We intend to create the domain-specific ORKG template for SWA research in such a way that it provides long-lasting value for prospective use w.r.t. the aforementioned properties: maintainability, usability (i.e., readability and understandability of templates for users), and suitability for the intended purpose (e.g., knowledge retrieval). Moreover, the specified template should fit to the design restrictions in the ORKG, supporting the general research knowledge description concept of research findings and their evidence in Section 3.1. In our context, we face three main conditions for the SWA data schema mapping to ORKG template: hierarchical structure (in depth and width), finite set of options (based on standards, a common community understanding, or expertise), and complex data structures.

For the tool-supported data extraction phase in the literature review, we clustered threats to validity and the replication artifacts to a superordinate category, namely *Validity*. This is also reflected in the first version of our ORKG template (cf. Figure 2) in a hierarchy of classes, fostering the structural depth of elements. However, an alternative could be to smooth the order of the elements, fostering width and less depth of the hierarchy. In future work, we have to investigate what implications this might have for machine-processable aspects, e.g., a retrieval approach using LLMs.

[3] https://gitlab.com/SoftwareArchitectureResearch/StateOfPractice/-/wikis/Data-Extraction/Taxonomy (last accessed on 10/17/2024)

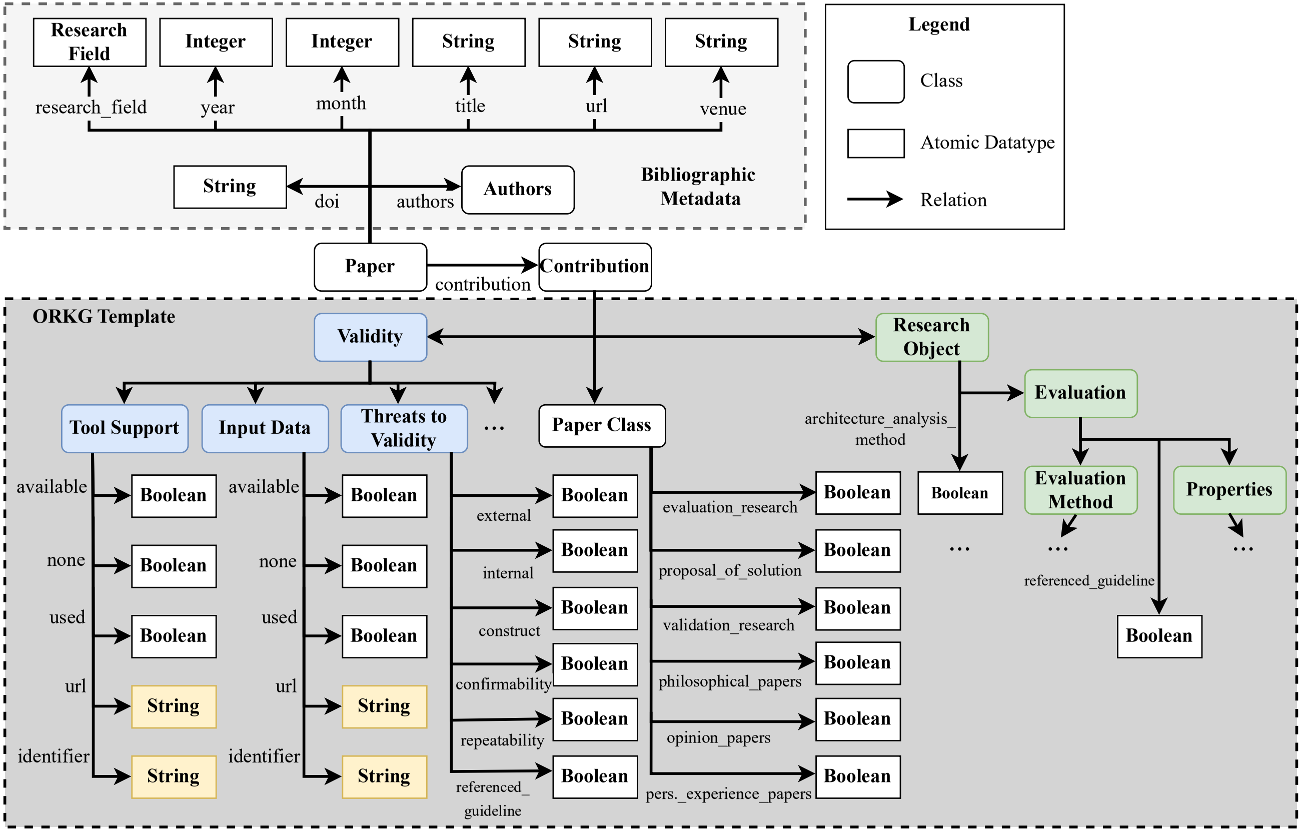

Fig. 2: ORKG template for software architecture research based on the data schema of [Ko22a] (short version 1.1); notation adapted from [Ka24]

Further, the elements of the data schema consist of multiple categorization types, each consisting of one or more values from a predefined and finite set of options. However, the concept of ORKG template does currently not provide such a datatype. Therefore, we identified two options for a multiple-choice concept in ORKG: (1) using plain text as String, and (2) using a list of boolean values. Both options come with advantages and drawbacks. First, using plain text as String would allow the users to flexibly input their choices. However, this option does not support restrictions on the input that is given in our data schema by expertise. In addition, it argues against the consistency in the data values as data types in the contribution that may relate to the same concept are not guaranteed to have the same value. This includes simple issues like casing, typos, synonyms, etc. Second, using a list of boolean values would force users who add new papers to ORKG with this template to select from predefined choices relevant to the specific research field. However, each choice would be a boolean to determine whether it is selected or not, reflecting a drawback of this approach since it is less flexible for future extensions. If the list is later updated to include more choices, all papers that have already used the template would have an empty value for the new choice. This requires a manual revision for each existing paper. In our case, we choose the most fitting option, depending on the property. Due to these considerations, we have to review the template in future maintenance.

Lastly, for complex data structures that are, e.g., reflected in a cyclic relationship and multidimensional cardinality w.r.t. the description of research findings through *Research Object*, *Evaluation Method*, and *Property*, we decided to specify a nested part in the ORKG template to adequately describe the knowledge without information loss (cf. Figure 2).

3.4 Visulite: Visualization of Literature

Besides having classes and atomic data types in text form, ORKG can integrate visualizations (of data like research results) in the templates, supporting multi-modal data while improving and enhancing research knowledge representations. In addition, the ORKG offers ORKG comparisons as one of its main features to provide a concise tabular overview of the state of the art on a specific research question, including visualizations. In the context of work on the data schema for SWA, Chebbi et al. developed a serverless application [CFK23a] to provide visualizations of the replication package [Ko22b], supporting various chart types (currently bar charts, pie charts, and bubble charts) to foster a better understanding and representation of the raw data results as well as a table with filtering options. Consequently, these generated visualizations can easily be integrated into ORKG services. Furthermore, Visulite ensures a flexible representation of the data according to the needs of the researchers. For instance, in addition to the aforementioned filter functionality, the user can select the data that he needs to be represented in the charts from a predefined list. The tabular representation may also be adjusted according to the user's wishes, e.g., by selecting the columns that should be shown, hiding any unnecessary information and minimizing the complexity of the resulting representation. In addition to the different visualization possibilities, the obtained results can be persisted in different ways: either by using the *.svg* files' extraction possibility for

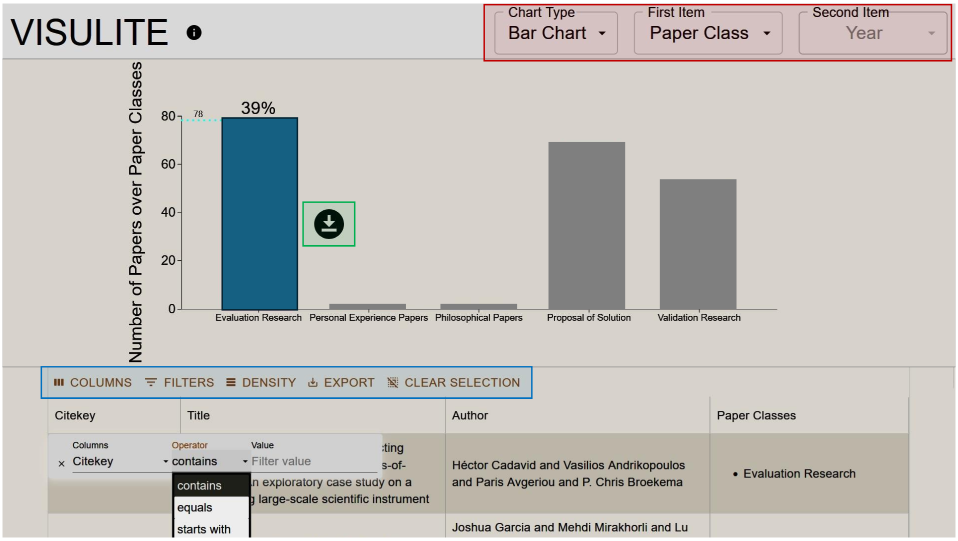

Chart Type: Bar Chart
First Item: Paper Class
Second Item: Year

Columns: Citekey
Operator: contains
Value: Filter value

Citekey	Title	Author	Paper Classes
	ting s-of- n exploratory case study on a g large-scale scientific instrument	Héctor Cadavid and Vasilios Andrikopoulos and Paris Avgeriou and P. Chris Broekema	• Evaluation Research
		Joshua Garcia and Mehdi Mirakhorli and Lu	

Fig. 3: Web interface of our tool Visulite. Colored boxes denote core tool features: settings for charts (red box), *.svg* download option (green box), and a table with filtering and export options (blue box).

the charts or the *.csv* files for the table's entries. Should the application's options not be sufficient to fulfill its users' requirements, the software architecture's conception enhances the extensibility and reuse of its components. By defining an internal model to represent the data and separating it from the input format and the resulting representations, users can introduce different parsers in addition to the existing *BibTex* parser. As a result, the application can be leveraged to support different input formats. Additionally, decoupling the data structure in the model from its representation in the charts allows for the reuse of the different existing charts and enables its extension with additional ones. In this regard, it is worth mentioning that the external library, d3.js, utilized for designing the charts, offers the developer more freedom during the design of the UI as it allows for the modification of even the smallest details. We encourage the research community to (re-) use our tool. For this, please refer to the corresponding code repository and the documentation [CFK23b].

4 Related Work

Following, we present related work concerning ORKG template developments in the research field of SE. Currently, we can identify another sub-research field in SE represented, in ORKG, namely requirements engineering (RE). The corresponding ORKG template is consequently developed to describe empirical research in RE, which currently covers the six themes *research paradigm*, *research design*, *research method*, *data collection*, *data analysis*, and *bibliographic metadata* [Ka23]. KG-EmpiRE represents the resulting community-maintainable KG. All supplementary materials are openly available [Ka24].

5 Conclusion

In this paper, we explored the capabilities of ORKG templates for semantic structuring in a sub-research field of software engineering (SE), namely software architecture (SWA) research. First, we presented our initial data schema for SWA research published at the flagship conference ICSA [Ko22a] that intended to provide an overview of the current state of practice in this research field. At the same time, we set an initial standard in this community for discussion and collaboration, describing research artifacts based on their findings and evidence. Second, we introduced the corresponding ORKG template and discussed design decisions in the transformation process from the data schema to the template. In this regard, we aim to provide hints for other researchers to construct a domain-specific template based on our experiences. Third, we presented our tool Visulite [CFK23a]. The tool contributes to graphical representations of raw data in replication packages, supporting multi-modal knowledge graph population and dashboards. In our long-term goal, we plan to document and maintain our artifacts (i.e., data schema, transformation process, and corresponding ORKG template) for the research communities. Our open-access repository captures these evolution scenarios and design decisions for comprehension, traceability, and benchmarking purposes. In future work, we aim to investigate and compare other ORKG templates in related research fields (e.g., requirements engineering; cf. Section 4) to foster consensus in the SE research community w.r.t. semantic structuring of scientific knowledge.

Acknowledgements

This work is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under the National Research Data Infrastructure – NFDI 52/1 – project numbers 501930651 (NFDIxCS), 442146713 (NFDI4Ing), 460234259 (NFDI4DataScience), 501865131 (NFDI4Energy), and supported by funding from the pilot program Core Informatics at KIT (KiKIT) of the Helmholtz Association (HGF).

References

[CFK23a] Chebbi, F.; Fuchß, D.; Kaplan, A.: Visulite (VISUalization of LITErature), 2023, URL: https://softwarearchitectureresearch.github.io/visulite/.

[CFK23b] Chebbi, F.; Fuchß, D.; Kaplan, A.: Visulite (VISUalization of LITErature) Dev and Docs, 2023, URL: https://github.com/SoftwareArchitectureResearch/visulite.

[DKJ05] Dybå, T.; Kitchenham, B. A.; Jørgensen, M.: Evidence-Based Software Engineering for Practitioners. IEEE Software 22 (1), 2005, DOI: 10.1109/MS.2005.6.

[FM10] Feldt, R.; Magazinius, A.: Validity Threats in Empirical Software Engineering Research - An Initial Survey. In: 22nd International Conference on Software Engineering & Knowledge Engineering. Knowledge Systems Institute Graduate School, 2010.

[GW16] Galster, M.; Weyns, D.: Empirical Research in Software Architecture: How Far have We Come? In: 2016 13th Working IEEE/IFIP Conference on Software Architecture (WICSA). 2016, DOI: 10.1109/WICSA.2016.10.

[GW23] Galster, M.; Weyns, D.: Empirical research in software architecture — Perceptions of the community. Journal of Systems and Software 202, 2023, doi: 10.1016/j.jss.2023.111684.

[IE90] IEEE: IEEE Standard Glossary of Software Engineering Terminology. IEEE Std 610.12-1990, pp. 1–84, 1990, doi: 10.1109/IEEESTD.1990.101064.

[IS11] ISO: Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models, Standard, International Organization for Standardization, 2011.

[Ja19] Jaradeh, M. Y.; Auer, S.; Prinz, M.; Kovtun, V.; Kismihók, G.; Stocker, M.: Open Research Knowledge Graph: Towards Machine Actionability in Scholarly Communication. ArXiv abs/1901.10816, 2019, url: https://api.semanticscholar.org/CorpusID:59413794.

[Ka23] Karras, O.; Wernlein, F.; Klünder, J.; Auer, S.: Divide and Conquer the EmpiRE: A Community-Maintainable Knowledge Graph of Empirical Research in Requirements Engineering. In: ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM). 2023, doi: 10.1109/ESEM56168.2023.10304795.

[KA24] KARAGEN (Knowledge Augmented Retrieval And Generation ENgine): Software Architecture Research / ORKG Template Structure, 2024, url: https://gitlab.com/software-engineering-meta-research/karagen/software-architecture-research/orkg-template-structure.

[Ka24] Karras, O.: Divide and Conquer the EmpiRE: A Community-Maintainable Knowledge Graph of Empirical Research in Requirements Engineering - A Sustainable Literature Review for Analyzing the State and Evolution of Empirical Research in Requirements Engineering, 2024, url: https://github.com/okarras/EmpiRE-Analysis.

[KDJ04] Kitchenham, B. A.; Dybå, T.; Jørgensen, M.: Evidence-Based Software Engineering. In: 26th International Conference on Software Engineering (ICSE 2004). IEEE Computer Society, pp. 273–281, 2004, doi: 10.1109/ICSE.2004.1317449.

[KKR22] Kaplan, A.; Kühn, T.; Reussner, R.: Unifying Classification Schemes for Software Engineering Meta-Research, 2022, arXiv: 2209.10491 [cs.SE].

[Ko22a] Konersmann, M.; Kaplan, A.; Kühn, T.; Heinrich, R.; Koziolek, A., et al.: Evaluation Methods and Replicability of Software Architecture Research Objects. In: 19th IEEE International Conference on Software Architecture, ICSA 2022. IEEE, pp. 157–168, 2022, doi: 10.1109/ICSA53651.2022.00023.

[Ko22b] Konersmann, M.; Kaplan, A.; Kühn, T.; Heinrich, R.; Koziolek, A., et al.: Replication Package of “Evaluation Methods and Replicability of Software Architecture Research Objects”. In: IEEE 19th ICSA 2022. 2022, doi: 10.1109/ICSA-C54293.2022.00021.

[KWH21] Kaplan, A.; Walter, M.; Heinrich, R.: A Classification for Managing Software Engineering Knowledge. In: EASE 2021: Evaluation and Assessment in Software Engineering. ACM, pp. 340–346, 2021, doi: 10.1145/3463274.3463357.

[Ra20] Ralph, P.; Baltes, S.; Bianculli, D.; Dittrich, Y.; Felderer, M., et al.: ACM SIGSOFT Empirical Standards. CoRR abs/2010.03525, 2020, arXiv: 2010.03525.

[RSS23] Rossenova, L.; Schubotz, M.; Shigapov, R.: The Case for a Common, Reusable Knowledge Graph Infrastructure for NFDI. In: 1st Conference on Research Data Infrastructure - Connecting Communities. TIB Open Publishing, 2023, doi: 10.52825/CORDI.V1I.266.

[TDV04] Taverniers, I.; De Loose, M.; Van Bockstaele, E.: Trends in quality in the analytical laboratory. II. Analytical method validation and quality assurance. TrAC Trends in Analytical Chemistry 23 (8), pp. 535–552, 2004, doi: 10.1016/j.trac.2004.04.001.