ONTOLOGY-BASED TOOLS FOR ARCHITECTURE ANALYSIS OF CYBER-PHYSICAL SYSTEMS

Liliana DOBRICA

UNST POLITEHNICA of Bucharest, Romania
liliana.dobrica@upb.ro

Abstract
This article focuses on the challenges regarding building ontologies and ontology-based tools with the purpose of using them in the architectural analysis of cyber-physical systems (CPSs). This research explores the main knowledge of CPSs architecting considering that the recent maturation of the existent standards about description, processes and evaluation of software intensive systems architecture can support better a systematic understanding and knowledge sharing to develop ontologies and smart tools based on ontologies. A tool that uses ontology knowledge aids the analyst to conduct a complex analysis method providing him/her more relevant and consistent guidance and more complete results. The ontology that formalize the consensus knowledge codified in standards and of the architecture analysis methods could be integrated in an application tool that becomes more trustable and efficient.
Keywords: knowledge sharing, standards, ontologies, software architecture, architecture analysis, cyber-physical systems

1. INTRODUCTION

Successful quality-based development and continuous evolution of cyber-physical systems (CPSs) depend on making right architectural choices (Agren et al., 2022)(Belomo, 2015). In the current context of the increasing complexity level and quality needs for CPSs, researchers and practicians revealed that it is very important to evaluate architectural choices to substantially identify and mitigate possible risks (Clements et al., 2002)(Ebert, 2015). The effort of the organizations responsible with the architectural evaluation (AE) process must be oriented to efficiently plan, execute and document these evaluations. A good knowledge management can support an organization to establish specific evaluation frameworks, methods and tools that can be used as the basis for multiple, recurring AE efforts (Babar et al., 2009) (Dobrica, 2021b) (Napoleao et al. 2021). An important challenge in this context is how can these organizations follow quality standards efficiently and effectively?

The selection of a suitable specific evaluation framework, method or tool could be facilitated when there is some expert domain knowledge guidance (Dobrica and Niemela, 2002) (Dobrica, 2011). Knowledge sharing and reuse has become an emerging topic of discussion and research in the field of knowledge management (Razavian et al., 2023). Existent solutions integrating reusable concepts for quality-driven architecture design of complex embedded systems (Dobrica, 2009) can be adapted to CPSs considering that CPS enhances the main features of complex embedded systems . Particularly for sharing and reuse of software architectural knowledge several open source based tools for have been proposed. Literature provides several solutions integrating detailed presentations and comparisons of the tools technical features (Ovaska et al., 2010). A better understanding of these solutions is when their comparison is based on a framework that reflects the coverage of the tools features.In this manner tools like WebOfPatterns (WOP), Stylebase and PAKME are discussed in (Hentonen and Mantilassi, 2009). Presently these solutions should be reconsidered for the new enhanced software architecture technologies integrated in the current development approaches of CPSs.

In recent years CPSs have gained significant attraction and have becoming increasingly omnipresent finding their applications in many and various domains including healthcare, energy, utilities, manufacturing and transportation. CPS integrates physical systems and computational entities that contain computing and communications cores (Lesch et al., 2023). These systems make intensive use of software and the physical and cybernetic components have complex interactions.The connection of physical and computational entities with behavior aspects is an important property of CPS and the quality is critical to every stakeholder (Ebert, 2015).

Management Research and Practice
Volume 15, Issue 3 / September 2023

ISSN
2067- 2462

As CPSs become more elaborated, their analyses tend to be more extensive. A CPS encompasses various elements of diverse types (i.e. software, human, hardware, and organizations) that are linked using different types of connections and networks. A smart tool used to analyse the quality of CPS architecture provides more appropriate guidance to analysists and enables better results. The analysis results can be improved using an ontology-based tool (Xu et al, 2019) (Carniel et al., 2023) for safety analysis. However, a comprehensive architecture evaluation process requires tools that are able to capture in an ontology formalism the complexity of the codified knowledge of standards documents.

CPSs are becoming more complex. Therefore, a CPS must follow regulations and recommendations of certification standards. CPS development process include quality analysis activities at early stages to discover and identify possible problems. Tools assistance and techniques are necessary to improve and ease the analysis. Ontology-based tools incorporate semantic knowledge achieving better results and automating the analysis. The knowledge domain of software engineering has considered the use of ontologies (Calero et al., 2010), but the CPS heterogenity requires new solutions based on a general common understanding of concepts and relations.

The content of this article presents the main elements that can be used to describe CPS structural architecture. It illustrates the presentation with a simple CPS example from the healthcare domain that includes an insulin pump. The next section focuses on the recent architecture standards of software intensive systems that provides a community's general common understanding of the concepts and relationships used to describe architecture representation, processes and the evaluation framework. It continues with a detailed discussion about tool assistance in CPS architectural analysis focusing on ontology development and ontology-based tools. It ends with conclusions regarding the benefits and drawbacks of using such tools and ontology knowledge representation.

2. CYBER-PHYSICAL SYSTEMS STRUCTURE

Many definitions of CPS have been recently captured from the existent literature by (Lesch et al., 2023). In a general view on a structural description of a CPS it can be reduced to a reunion of cyber and physical entities, and the links connecting these entities. However, CPSs make intensive use of software, employ processors with increasing processing power, networks of multiple sensors and actuators. Processors and different networks support the cyber components of a CPS to analyze the data monitored from the physical components by using complex algorithms. The results of the algorithms are actions to control and optimize the behavior of the physical entities.

A simple example from the healthcare domain that is classified as a CPS with hierarchical control system having the human in the loop is the intelligent insulin pump system as described in (Carniel et al., 2023) and presented in Figure 1. In this example CPS integrates patient, smartphone, insulin pump and a glucose sensor. The glucose sensor is used to identify the glucose level form the patient body. This is a hierarchical control system, since the patient controls the smartphone, and the smartphone controls the actuator, specifically the insulin pump. Through a Bluetooth connection, the smartphone receives data and status information from the insulin pump controller and glucose sensor, processes data, and issues command to the insulin pump controller. In addition, the smartphone can contact a medical service in an emergency case, and receive a response from the medical service. The smartphone application is downloaded and updated by a virtual store and it is responsible for storing the information in a local database. The insulin pump is placed in the patient's belly, and the glucose sensor is placed in the patient's arm. The glucose is measured at five-minute intervals. The status of the measured values is communicated via Bluetooth with a smartphone. The smartphone uses the received information and knowledge of the chronical disease, diabetes, and controls the pump.

There is an interaction between the smartphone and the medical service. In an emergency situation the smartphone communicates with the medical service and requests help from the medical center. The smartphone can communicate in an autonomous way, when the patient is unconscious. If the patient becomes unconscious, the smartphone requests an ambulance from the medical service. In this scenario, it is the responsibility of the smartphone to send a request message.

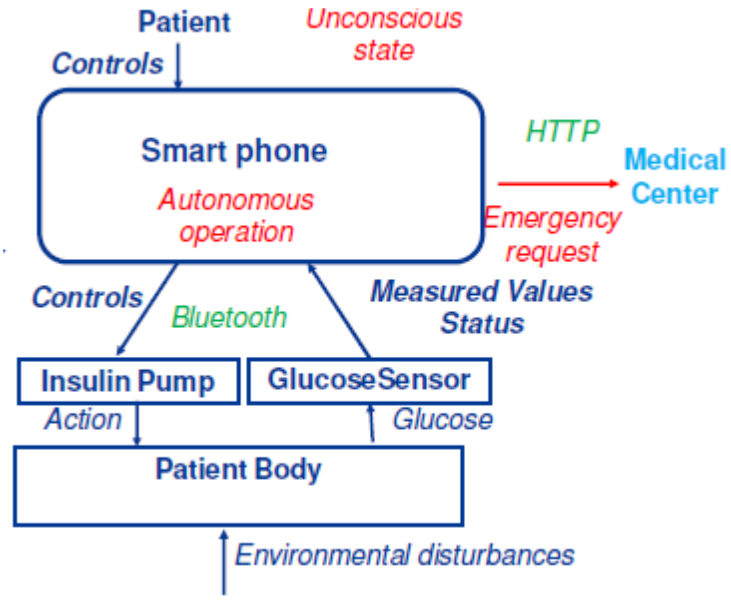

FIGURE 1 - EXAMPLE INSULIN PUMP CPS.

Due to the maturation of architecture representation technology, a design from scratch for a software intensive system doesn't have to be created. (Carniel et al., 2023) recently synthesized a generic structural diagram of a CPS that is shown in Figure 2 as a description of the main types of elements and types of links between these elements.

Figure 2 illustrates in the structural diagram of a CPS the specific types elements that can be used to build it. It can be mentioned controller, control actions, actuators, sensors, feedback, input, output, external information.

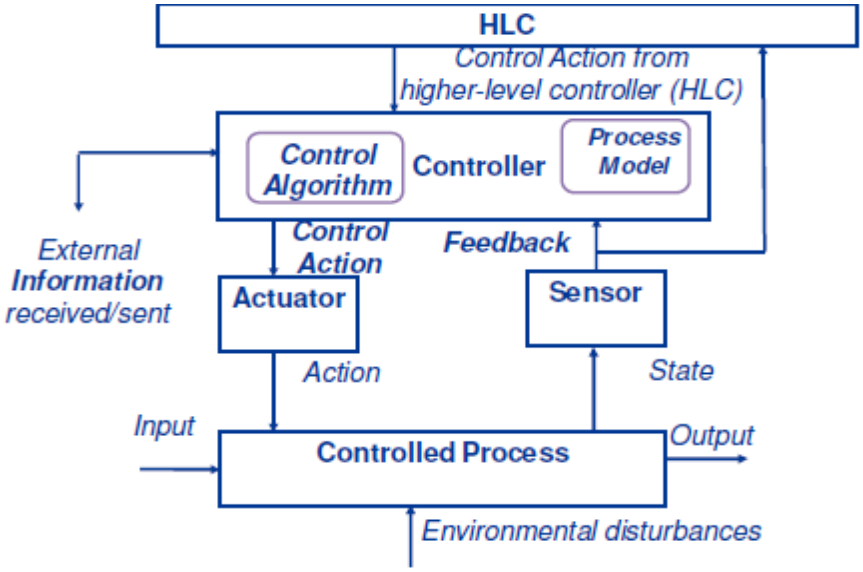

FIGURE 2 - GENERIC STRUCTURAL DIAGRAM OF CPS (ADAPTED FROM [1]).

The controller uses an actuator to interact with the controlled process and receives the feedback through a sensor.

Each interaction between elements has a specific link. There could be considered several types of links, including link for control actions, link for feedback, or link for information. A link for control actions: is used to represent a flow of control action issued by a controller or a higher-level controller. A link for feedback is used to represent a flow of feedback from a sensor or controlled process to a controller. In addition, it can also be used to send feedback from the controller to a higher-level controller. A link for information is used by a controller to send/receive information to/from an external system.

Management Research and Practice
Volume 15, Issue 3 / September 2023

3. STANDARDS FOR CONSENSUS OF SOFTWARE ARCHITECTURE CODIFICATION KNOWLEDGE

Software architecture technology has benefited of international organizations converging support and efforts to provide a coherent standard framework since this became a mature technology and ready for popularization. During its evolution through several phases from the basic research to ready for popularization many researchers contributed to its growth (Shaw and Clements, 206) (Shaw and Clements, 2009). Significant contributions in the software architecture analysis and evaluation is prezented in (Dobrica and Niemela, 2002) and many more studies after this publication have explored the development of new or enhancement of existent solutions.

One remarkable signal of a production-ready technology is the good standards. In 2000 it has been created IEEE 1471, the short name for the first formally standard known as ANSI/IEEE 1471-2000, *Recommended Practice for Architecture Description of Software-Intensive Systems.* (Maier et al., 2001) explained the approval process for a consensus on good architectural description practices. It was the least normative of IEEE standards and from this, in 2007, ISO/IEC adopted the ISO/IEC/IEEE 420110:2007 "*Systems and Software Engineering -- Recommended practice for architectural description of software-intensive systems".* During the golden age of software architecture community more standardization efforts have contributed to create more consensus documents for a better definition and understanding of architecture description, architecture processes and architecture evaluation framework.

Newer standards are emerging all the time to codify the current best practices and insights of both systems and software engineering communities. The most recently ones are (ISO/IEC/IEEE 42010,2022) Software, systems and enterprise – Architecture description, (ISO/IEC/IEEE 42020,2019) - Software, systems and enterprise- Architecture processes, and (ISO/IEC/IEEE 42030:2019) Software, systems and enterprise – Architecture evaluation (AE) framework. These documments codify existing consensus on key concepts and terminology providing a foundation that organization can use for their own industry segment.

The AE framework focusses on three AE tiers, namely, evaluation synthesis (AE_ES), value assessment (AE_VA) and architecture analysis (AE_AA). Each tier has specific elements . AE_ES has objectives, approaches, factors, and results. AE_VA has objectives, methods, factors, and results. Finally, AE_AA has objectives, methods, factors and results. Approaches and methods are typically knowledge elements that are reused across applicable contexts.

Considering the AE_AA tier it is important to remark that it examines the key attributes of an architecture and the impact on stakeholders or on the environment. Also is good noting that AE_AA examines architecture vision, principles, concepts and properties that are relevent to achieving the AE_AA objectives. In practice, AE_AA looks at aspects such as forms, patterns, structures, behaviors, functions, flows. In general AE_AA examines the assumptions about technologies, operations, policies, constraints, etc.

Using several examples, the AE standard clarifies the difference between terms such as assesment and analysis. The assessment is used to determine the extent to which an architecture provides value to stakeholders and the goal is oriented towards fundamental objectives, while the analysis goal focuses on the means objectives that are often multi-level and their basis of work is technical.

From the general AE framework specific frameworks targeting various industry segments (i.e. automotive), common business processes (i.e. portofolio management, common business architectures (banking, telecom, travel, etc) can be derived. These specific frameworks increase efficiency with which architectures can be evaluated due to capturing and reusing of common concepts. A well-known example of a specific AE framework is ATAM (Dobrica and Niemela, 2002)(Clements et. Al, 2002). ATAM includes an evaluation approach based on the use of quality attribute workshops. ATAM specifies both value assessment and quality attribute analysis method to be used.

TABLE 1 - THE RECENT STANDARDS RELATED TO ARCHITECTURE KNOWLEDGE CODIFICATION

Standard	Scope
ISO/IEC/IEEE 42010:2022	Specifies requirements for the structure and expressin of an architecture description for various entites
ISO/IEC/IEEE 42020	Establish a set of process descriptions for the governance and management of a collection of architectures.
ISO/IEC/IEEE 42030	Specifies means to organize and record architecture evaluations (AE).

Suppliers for software-intensive systems as well as component suppliers must comply with formal concepts from standards such as those summarized in Table 1. These have the same origins in the software architecture community, but address different challenges from description, architecting processes and a more detailed information regarding architecture evaluation (AE) . Nevertheless, it can be seen that value and quality are related to factors used in the AE effort. The evaluation factors decomposition can be performed in a strict hierachical fashion. Also the objectives in each tier can be structured in a way where objectives in one tier will satisfy objectives in an upper tier. However this direct correlation of objectives is not always possible. AE factors are similar to quality attributes in ATAM or quality characteristics in the ISO/IEC 25000 family of standards on Systems and Software Quality Requirements and Evaluation (Nakai et al., 2016).

AE standard gives detailed information regarding the value concept and its importance during architecting, value-focused thinking, value models and value measures. It is mentioned here that the key characteristics of system architecting is to maximaze stakeholder value and some common approaches to assess the value of an engineered system are presented in the table bellow (Table 2):

TABLE 2 - COMMON APPROACHES TO ASSES THE VALUE OF AN ENGINEERED CPS

Simulation models to simulate every function and its interaction with the surrounding context to calculate the value of the system of interest;
Simulation experiments to quantify the value of a certain dataset (ex. numerical data prediction)
Rating the importance of different customer attributes and their coupling to design features by means of QFD (quality function deployment)
Computing the net present value of an architecture to determine its cardinal ranking amongst collection. Adopting the value of informatiom approach to identify probabilities of different scenarios and availability of useful information in these scenarios to make a data-based decision.
Comparing the capabilities of the system architecture with the stakeholder requirements and use rule-based expert systems to assess the value of the system architecture.

A ring model of value is described in the AE standard. This is useful in delivery value to stakeholders and continously manage and upgrade long-life systems. The value model has three levels that are organized to focus on th following aspects:

the system to be created (system envisioned, system designed and architecture articulated, components specified-developed-assembled, system tested operational readiness and system activated),

the system's purpose (intervention strategy, purpose identified, operational results and purpose known) and

the value of the system to its stakeholders (societal values, problem discerned, problem understood, solution effect envisioned, context adapted to system, effects on problem observed, value of system quantified).

AE standard defines stakeholders values, qualities and measures. Quality and value measures are quantifiable (i.e. more user-friendly, less reliable, enhanced performance, etc) and this quantification helps address stakeholders needs appropriately.

Management Research and Practice
Volume 15, Issue 3 / September 2023

The method for tracing value from conception to realization in an assured manner is the value articulation framework. In the context of technology management the essential ingredients in establishing this trace are stakeholder perception of value, quality characteristics of the technology that is developed, technology management discipline and the value that accrues to the given stakeholder.

Providing information on quality attributes, quality models and quality measures for a quality concept during architecting focuses on a pragmatic interpretation as non-inferiority or superiority to something for quality in business, engineering and manufacturing.

The architecture quality attributes (QAs) are defined as the the extent to which the architecture can deliver value to its stakeholders. These are essential and distinguishing attributes that have a pragmatic interpretation of the architecture's inferiority or superiority. The architecture QAs are the overall factors that affect behavior, structure, design and experience of architecture and represent areas of concern that potentially impact the structure and behavior exhibited by the realized system. The extent to which the architecture handles a combination of QAs indicates the success of the architecting effort and overall quality of the realized system. Each QA has a taxonomy defined by measures, factors and methods. Measures are the parameters by which the attributes area measured. Factors are policies and mechanisms of the system and its environment that impact the stakeholder concerns. Methods are techniques for addressing concerns and processes for realizing the QAs during productions.

It is important that while conceptualizing architecture to address the architecture QA, it is necessary to consider potential impact of each QA on other stakeholder concerns. Trade-off analysis techniques aid architects in prioritizing QA. Architectural tactics describe how specific QA can be achieved. The importance of each QA depends on the context and the stakeholders concerns for which the specific architecture is conceptualized. A list of QAs is provided. Among the elements of the QAs list are enumerated *coherence, variability, commonality, flexible, verifiable, traceable*.

4. TOOLS ASSISTANCE IN ARCHITECTURAL ANALYSIS OF CPS

The architecture analysis process can be performed without tool assistance**,** but the results depend on the analyst's expertise. For instance, in the case of scenario-based analysis methods (Dobrica and Niemela, 2002) the analyst has to consider all the stakeholders' scenarios (i.e. scenarios elicitation), that could be pertinent or non-pertinent, resulting in wasting of time and increasing the possibility of making mistakes.

The analysis process where the analyst is assisted with normal tools might consider the use of spreadsheet-based tools or use of a general tool allowing the creation of meaningless connections between CPS types of elements or where CPS elements have an ambiguous type (an element is actuator and controller at the same time). Also it might be possible to use a tool that does not check what connections/links are in the structure or hierarchical structure.

The analysis process with a smart tool provides a systematic analysis. The analysis is performed in a more disciplined way. It provides automated analyses and efficient-time performance, aid the analyst by providing a pertinent guidance with valuable suggestions/ advice/, by providing recommendation or accurate information aimed at performing the analyses, by avoiding doubts and allowing a complete and more detailed analysis.

An ontology-based tool uses the ontology formal models to acquire and codify knowledge, makes inferences and saves the results (Carniel et al., 2023) (Dobrica, 2021a). The development of today's CPS needs systems theory-based approaches to identify recommendations and requirements and they need systematic techniques that can be employed in tools. Studies demonstrated that even for small systems it is easy to miss some important aspects. An ontology-based approach is a more trustable alternative to improve how to address such problems.

Such project needs to plan activities involved in the quality attribute ontology and the tool. The ontology represents a complete analysis knowledge and need to include all concepts of the method and their relations. Also the ontology should include a complete control structure such as controllers, process model variables, links between controllers an high-level controllers, and links between controllers and external systems. Also

the ontology has to consider the controlled process, its inputs, its outputs and environmental distubances. The relations are specified to allow inferences.

The ontology-based tool has a user interface with a precise control flow to conduct the analysis based on the analysis method. It provides a report that shows the complete results of the analysis. It uses asserted knowledge and inferred knowledge**.** The inferred knowledge is related to the inferences that can be made using the reasoner. The reasoner rearranges, infers, and classifies all the classes according to the ontology's relations and shows the subclasses knowledge. The inferred knowledge is read by the tool and is used to discover elements in the CPS structure and to verify what connections can be made by the elements to prevent mistakes and wrong connections.

Ontology-based tool trade-off features between a Web-based collaborative features that allows several users to access the same analysis concurrently and a desktop application features. A desktop version is an application that can be preferred to keep the tool performance because the ontology reasoner may consume the processor. An important benefit of formalizing codified knowledge in ontologies is that the ontology is verifiable. Experts can verify the proposed ontology, thus there is a confidence in using the smart tool. It is possible to verify if the tool respects the analysis method, for instance, if the tool considers all the method elements and their relations.

Ontology consists of structuring knowledge to make it accessible to a computer. It is a way to conceptualize a domain knowledge and the knowledge representation can be used by computers and humans. An ontology has a hierarchy of defined concepts, classes and subclasses, axioms, relations, properties, formal constraints, categories, classifications. It has an asserted hierarchy and produces an inferred hierarchy that is obtained by performing logical inferences with a reasoner, particularly the (HermiT or Pellet) reasoner are publicly-available OWL reasoners determine if the ontology is consistent and identify subsumption relationships between classes.

The ontology development tool, Protégé, is open-source. The approach follows the analysis method to elaborate the ontology. The architecture analysis method is composed of a sequence of activities. For each activity is created an ontology. The ontology of each activity is merged with the method ontology. The consistency of the method ontology is analyzed. One way to evaluate the ontology is to use the competence questions to verify if the ontology is able to provide the needed knowledge. Use of SPARQL queries to verify the expected responses and to infer that the ontology completeness is verified.

5. CONCLUSIONS

The standards are rather abstract and don't give much concrete guidance on how to efficiently use the framework into existing architecting processes. To provide more guidance, recently standard evolution tends to provide many prescriptive details on how to treat architecting processes, quality attributes and their traceability and validation, tool use, and QAs cases documentation and maintenance. The risk with this approach is that engineers/analysts and their management are overwhelmed and just follow what's described while not designing according to the QA needs, actual risks, and economic trade-offs.

Architectural analysis methods are lengthy, complex and requires a planned effort from the organizations. However, methods have steps to be executed in a sequential and iterative way with refinements. The ontology models the method analysis knowledge. The concepts involved in the analysis method should be domain specific and conform to AE framework. The knowledge and information obtained with the analysis are then used to design the CPS being analysed. In the design, other models such as UML, SysML or MARTE diagrams, are obtained. In the design phase more detailed information are then specified.

An important feature of ontology is that it is verifiable. Experts can verify if the proposed ontology contains all the relations, properties, formal names, categories, classifications, defnitions, and how its elements relate among them in the domain of the qality attribute analysis. The tool that employs an ontology can be verified if it respects all the elements and their relations. This property gives confidence in using this tool.

This exploration research is in progress. The main findings presented in this paper illustrate that an ontology-based smart tool could provide verification of the CPS structure diagrams. The smart tool could provide guidance support to the analyst. The use of ontology results in a more trustable smart tool. Nevertheless

ontologies allow formalizing heterogeneous domain knowledge. Considering this it is possible in the future work to create an ontology to formalize the analysis knowledge for distinct analysis methods for the same quality attribute.

An ontology-based approach that support the process of the standard AE framework through a smart tool is presented in this article. This research provide two important results. The first one is the ontology that represents the knowledge of an analysis method, resulting in an ontology with several concepts and relations. This knowledge is useful when complex analysis methods are considered. An ontology-based tool provides useful guidance. The ontology provides the knowledge of the quality attribute analysis that the tool uses to guide the analyst and produce consistent recommendations and more complete analysis. The tool suggests only the pertinent recommendations, which saves time for the analysis and reduces the possibility of errors.

A future challenge for the ontology model can be to consider integrating concepts from multiple quality attributes including the critical ones such as safety and security. A safety analysis ontology model can help in the identification of hazards and losses and a security analysis ontology model focuses on a threat model of cybersecurity threats. Moreover, ontology can integrate the knowledge of different analysis methods of the same quality attribute. An ontology model to analyze conflicts and strengthening between requirements and between the mechanisms identified to meet the requirements. It could be consider the usability of a tool with analysis of the strong points and weak points. The ontology can be modeled to store information about the performed analyses and this information to be the reused knowledge in other analyses.

REFERENCES

Agren, M. S., Knauss, E., Heldal, R., Pellicione, P., Alminger, A., Antonsson, M., Karlkvist, T., & Lindeborg, A. (2022). Architecture evaluation in continuous development, *Journal of Systems and Software*,. *Vol.184*,. Article 111111. https://doi.org/10.1016/j.jss.2021.111111

Babar, A., Dingsor, L., & van Vliet, H. (2009). *Software Architecture Knowledge Management Theory and Practice*, Springer.

Bellomo, S., Gorton, I., & Kazman, R. (2015). Toward Agile Architecture- Insights from 15 years of ATAM Data, *IEEE Software*, *32*(5), 38-45. https://doi.org/10.1109/MS.2015.35

Calero, C., Ruiz, F., & Piattini, M. (2010). *Ontologies for Software Engineering and Software Technologies*, Springer.

Carniel, A., de Melo, B. J., & Masaki, H. C. (2023). An Ontology-based Approach to Aid STPA Analysis, *IEEE Access*, *11*, 12677-12697. https://doi.org/10.1109/ACCESS.2023.3242642

Clements, P., Kazman, R., & Klein, M. (2002). *Evaluating Software Architectures*, Addison Wesley.

Dobrica, L., & Niemelä, E. (2002). A survey on software architecture analysis methods, *IEEE Transactions on Software Engineering* , 28(7), 638-653. https://doi.org/10.1109/TSE.2002.1019479

Dobrica, L. (2009). Integrating Reusable Concepts into a Reference Architecture Design of Complex Embedded Systems, *ICINCO 2009: Proceedings of the 6th International Confeference on Control, Automation and Robotics, Vol.3*, 234-237.

Dobrica, L. (2011). Exploring Approaches of Integration Software Architecture Modeling With Quality Analysis Models, *2011 Ninth Working IEEE/IFIP Conference on Software Architecture (WICSA)*, IEEE Computer Society, 113-122. https://doi.org/ 10.1109/WICSA.2011.23

Dobrica, L. (2021). A Study On Practical Approaches In Building Domain Ontologies, *Management Research and Practice*, *13*(4), 41-50.

Dobrica, L. (2021). Knowledge In Action Towards Better Knowledge Management In Organizations, *Management Research and Practice, 13*(2), 26-35.

Ebert, C. (2015). Implementing Functional Safety. *IEEE Software*, *32*(5), 84-89. https://doi.org/10.1109/MS.2015.126

ISO/IEC/IEEE 42010. (2022). IEEE/ISO/IEC International Standard for Software, systems and enterprise--Architecture description, in *ISO/IEC/IEEE 42010:2022(E)*, pp.1-74, 7 Nov. 2022, https://doi.org/10.1109/IEEESTD.2022.9938446

ISO/IEC/IEEE 42020. (2019). ISO/IEC/IEEE International Standard - Software, systems and enterprise -- Architecture processes, in *ISO/IEC/IEEE 42020:2019(E)*, pp.1-126, 19 July 2019. https://doi.org/10.1109/IEEESTD.2019.8767004

ISO/IEC/IEEE 42030. (2019). ISO/IEC/IEEE International Standard - Software, systems and enterprise -- Architecture evaluation framework, in *ISO/IEC/IEEE 42030:2019(E)*, pp.1-88, 19 July 2019, https://doi.org/10.1109/IEEESTD.2019.8767001.

Henttonen, K., & Matilassi, M. (2009). Open Source Based Tools for Sharing and Reuse of Software Architectural Knowledge, *2009 Joint Working IEEE/IFIP Conference on Sosftware Architecture and European Conference on Software Architecture*, 41-50. https://doi.org/10.1109/WICSA.2009.5290790

Hu, X., Liu, J., & Wang, Y. (2019). Multi ontology-based System-Level Software Fuzzy FMEA Method, *6th International Conference on Dependable Systems and Their Applications (DSA)*, 49-59. https://doi.org/10.1109/DSA.2019.00015

Lesch, V., Zulfe, M., Bauer, A., Ifflander, L., Krupitzer, C., & Kounev, S. (2023). A literature review of IoT and CPS – What they are, and what they are not, *Journal of Systems and Software, 200*, June 2023, 111631. https://doi.org/11.1016/j.jss.2023.111631

Maier, M., Emery, D.,& Hilliard, H. (2001) Software Architecture: Introducing IEEE Standard 1471, *Computer*, *34*(4), 107-109. https://doi.org/10.1109/2.917550

Nakai, H., Tsuda, N., Honda, K., Washizaki, H., & Fukazawa, Y. (2016), A SQuaRE-based Software Quality Evaluation Framework and its Case Study, *Proceedings of the 2016 IEEE Region 10 Conference TENCON*, 3704-3707.

Napoleao, B.M., de Souza, E. F., Ruiz, G. A, Felizardo, K.,R., Meinerz, G.V., & Vijaykumar, N. L. (2021). Synthesizing researches on Knowledge Management and Agile Software Development using Meta-ethnography method, *Journal of Systems and Software*, *178*, 110973. https://doi.org/10.1016/j.jss.2021.110973

Ovaska, E., Evesti, A., Henttonen, T., Palvianen, M., & Aho, P. (2010). Knowledge based quality-driven architecture design and evaluation, *Information and Software Technology*, *52*(6), 577-601. https://doi.org/10.1016/j.infsof.2009.11.008

Razavian, M., Paech, B., & Tang, A. (2023). The vision of on-demand architectural knowledge systems as a decision making companion?, *Journal of Systems and Software, 198*, 1111560. https://doi.org/10.1016/j.jss.2022.111560

Shaw, M., & Clements, P. (2006). The golden age of software architecture, *IEEE Software*, *23*(2), 31-39. https://doi.org/10.1109/MS.2006.58

Shaw, M., & Clements, P. (2009). "The golden age of software architecture" revisited, *IEEE Software*, 26(4), 70-72. https://doi.org/ 10.1109/MS.2009.83