Architectural Decisions in AI-based Systems: An Ontological View

Xavier Franch[0000-0001-9733-8830], Silverio Martínez-Fernández[0000-0001-9928-133X], Claudia P. Ayala[0000-0002-6262-3698] and Cristina Gómez[0000-0002-3872-0439]

Universitat Politècnica de Catalunya, Spain
{xavier.franch | silverio.martinez | claudia.ayala | cristina.gomez}@upc.edu

Abstract. Architecting AI-based systems entails making some decisions that are particular to this type of systems. Therefore, it becomes necessary to gather all necessary knowledge to inform such decisions, and to articulate this knowledge in a form that facilitates knowledge transfer among different AI projects. In this exploratory paper, we first present the main results of a literature survey in the area, and then we propose a preliminary ontology for architectural decision making, which we exemplify using a subset of the papers selected in the literature review. In the discussion, we remark on the variety of decision types and system contexts, highlighting the need to further investigate the current state of research and practice in this area. Besides, we summarize our plans to move along this research area by widening the literature review and incorporating more AI-related concepts to this first version of the ontology.

Keywords: AI-based systems, Software architecture, Architectural decisions, Ontologies, Quality attributes, Architectural Views, UML class diagrams.

1 Introduction

The conception, development and deployment of software systems that embed artificial intelligence (AI), what we call *AI-based systems*, has become commonplace in the last decade. This is mostly due to the increased computer processing power, availability of larger datasets, and constant formulation of better AI algorithms which has advanced the AI field to unprecedented levels of adoption [2]. Classical software engineering disciplines have been used to produce AI-based systems, from requirements engineering to testing, remarkably including software design principles, methods and techniques to deliver software architectures for AI-based systems [17].

Given that an AI-based system is nothing else than a particular type of software system, it can be thought that the whole discipline of software design and software architectures apply. However, the literature has reported significant challenges that are particular to architecting AI-based systems, related to design principles, design quality and software structure [17]. As a response to these challenges, a number of research approaches have formulated design strategies to cope with specific quality attributes and concrete AI infrastructure proposals [23]. However, these research approaches take a pragmatic perspective, focusing more on resolving the problem at hand rather than considering the particular problem as an instance of a more generic situation. This fact hinders knowledge transfer from one experience to another and makes it difficult to decide whether a solution formulated in one paper applies to a new problem.

In order to overcome this challenge, in this exploratory paper we present our ongoing research towards the formulation of a unifying conceptual framework aimed at defining the concepts that characterise the process of architectural decision-making [22]. Through a literature review upon a set of 41 papers in the field of software design and software architectures in AI-based systems, we extract the main concepts relevant to architectural decisions for this type of systems, and propose a preliminary ontology that captures the knowledge that is relevant to that process. We finalise the paper with a research agenda for this line of investigation.

2 Background

2.1 Architecting AI-based systems

Software design and architecture of AI-based systems, like other software development activities, differs on AI-based systems with respect to traditional software systems. As a result, there has recently been emerging research on software architecture for AI-based systems, as well as dedicated events (e.g., CAIN@ICSE, SAML@ECSA).

Two of the most studied topics have been design strategies to cope with specific quality attributes (e.g., classical attributes such as safety and reliability [11], or emerging attributes such as energy efficiency [7]), and AI infrastructure proposals (e.g., for sharing models as microservices) [17]. Serban et al. argue that traditional software architecture challenges (e.g., component coupling) also play an important role when using AI components; along with new AI specific challenges (e.g., the need for continuous retraining) [20]. They establish a link between architectural solutions and software quality attributes, to provide twenty architectural tactics used to satisfy individual quality requirements of systems with ML components. Furthermore, Yokoyama et al. have studied architectural patterns for AI systems [25].

2.2 Architectural Decisions

Architectural decision-making is a well-established research area in the field of software architecture. In a recent semi-systematic literature review, Bhat et al. report over 250 publications on the area, with a clear increase from the year 2005 [3]. Research proposals can be arranged according to several dimensions, mainly: 1) what are the drivers that influence architectural decisions (e.g., quality attributes [1]), 2) in which type of system architectural decisions apply (e.g., microservice APIs [26]).

In our paper, we are interested in the study of architectural decisions on AI-based systems. Studies in this area are scarce. A notable exception is the work by Warnett et al., which provides initial industrial evidence of architectural decisions faced by practitioners when designing an ML pipeline [24]. While the information provided in this paper is really valuable, it is focused in one particular AI context (ML pipelines) and does not articulate the gathered knowledge into a comprehensive framework, which is our final aim in this exploratory paper.

3 Research Questions and Method

The purpose of this paper is to present the main ideas, current research and future agenda for our research on AI-based system architectural decision making. To this aim, we formulate the following research questions:

RQ1. *What are the concepts that influence architectural decisions in AI-based systems?*

RQ2. *How can we specify a conceptual framework for these concepts and decisions?*

With RQ1, we want to elicit and characterise the concepts that need to be considered when designing AI-based systems. We expect these factors to be related to classical quality attributes such as time efficiency or accuracy [11], but also to emergent concerns, e.g. related to green AI [19].

To answer this research question, we use the result of a recent systematic mapping study on software engineering practices for AI-based systems [17], which includes software design as one of the SWEBOK knowledge areas. In addition, we consider the contributions presented in two recent venues, namely the 1st International Workshop on Software Architecture and Machine Learning[1] (SAML) and the March 2022 special issue on AI and software engineering published in IEEE Computer[2]. We analysed the resulting 41 papers (34 papers from [17] related to software design, and 5 papers from the SAML workshop and 2 architecture-related papers in the IEEE Computer issue) and extracted relevant information that we use to respond to RQ1. In more detail: *(i)* we split the 41 papers among the four authors at approximately equal share; *(ii)* every author read and extracted data of the papers assigned to them; *(iii)* we met at weekly basis and commented the result of the work in that week, consolidating the analysis and converging into a shared understanding; *(iv)* as we advanced, we synthesised the result in a data extraction form represented with a spreadsheet. It is worth noting that some of these 41 papers do not present concrete proposals because they are empirical studies reporting current research or practice; therefore, we were not able to extract information related to RQ1 from them.

For answering RQ2, we synthesised the knowledge gained from this extracted information using an ontology to present the concepts and their relationships. Ontologies are a widely used artefact used for knowledge representation and management [13], which is the primary goal of our work. In the area of architectural decisions, ontologies are widely used to represent architectural knowledge [18, 10]. Therefore, it seems a natural choice for our goal. In this exploratory paper, we represent ontologies in a lightweight form, using UML class diagrams [14] and a glossary of terms. We define the terms relying on standards and former papers as much as possible, although in some cases we have opted by providing our own definitions, better aligned to the pursued objective of the paper.

[1] https://saml2021.disim.univaq.it/

[2] https://www.computer.org/csdl/magazine/co/2022/03

4 An Ontology for AI-based Systems Architectural Decision-Making

Table 1 compiles the main concepts emerging from our analysis, and Figure 1 presents a class diagram relating these concepts. Central to the ontology is the basic concept of *Architectural Decision*, which we adopt from De Boer *et al.* [4]. Important to this definition is the understanding that an architectural decision may call for the need of *subsequent* architectural decisions. Architectural decisions can be classified into one *Decision Type* and may be constrained in their applicability to one or more *Contexts*. We do not impose a closed enumeration of neither decision types nor contexts; instead we foresee that these types will naturally emerge as the knowledge on AI-based systems architecture grows.

Table 1. Concepts of the ontology.

Concept	Definition	Source
Architectural Decision	Decision that is assumed to influence the architectural design of an AI-based system and can be enforced upon this architectural design, possibly leading to new concerns that result in a need for taking subsequent decisions	[4]
Decision Type	A type in which an architectural decision may be classified	From authors
Context	Any information related to an AI-based system that can be used to characterise the applicability of an architectural decision	Adapted and simplified from [9]
Quality Attribute	Measurable physical or abstract property of an AI-based system that bears on its ability to satisfy stated and implied needs	Adapted from [11]
Impact	The degree in which an architectural decision relates to a quality attribute	From authors
Architectural Element	Any type of element that can appear in an architecture, either an abstract concept (e.g., an architectural style) or some binary object (e.g., a software component or a data file)	From authors
AI-related Architectural Element	A class of Architectural Element that embeds or represents AI knowledge, e.g. an ML model, an implemented AI algorithm or a dataset	From authors
Architectural View	Representation of the whole system from the perspective of a related set of concerns	From [12]

Architectural decisions are taken according to their *impact* on a number of *Quality Attributes* that are considered relevant for the AI-based system under development. To measure the impact of an architectural decision, we use the qualitative scale proposed in the iStar language [8] with four scales ranging from strong positive influence ("make") to strong negative influence ("break"). On the other hand, an architectural decision affects a number of *Architectural Elements* (at least one), which can eventually be *AI-based Architectural Elements*, typically embedding some ML model or offering some AI algorithm (maybe in the form of a library) or even a dataset. Components are related to a particular *Architectural View*, since architectural decisions may be made at different levels of abstraction. Types of architectural elements and views are left open, subject to further investigation.

The ontology recognizes the hierarchical nature of architectural decisions, architectural elements and quality attributes by means of recursive many-to-many associations in the class diagram.

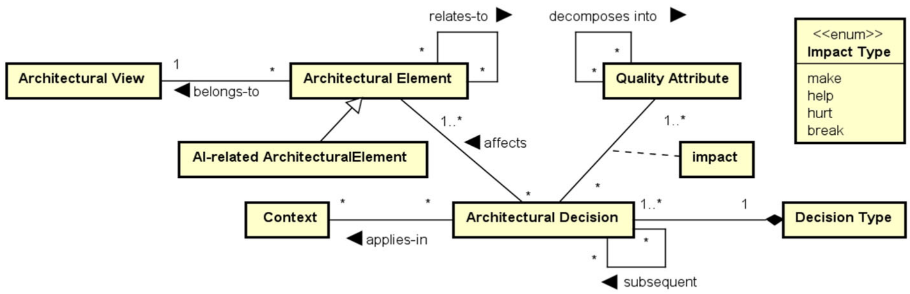

Fig. 1. Class diagram relating all concepts of the ontology

Table 2 exemplifies the conceptual framework on a subset of the 41 selected papers, by giving values to the concepts represented in the class diagram. We describe in more detail three particular cases below.

Example 1. Kumar et al. [16] propose an AI-system to decide the best location of chargers for electric vehicles based on spatiotemporal data from citizens' vehicles. Of course, this raises privacy concerns. The paper wants to exploit the fact that vehicles currently have enough computational power to train AI models.

What we learnt. Kumar et al. start the architectural decision-making process by applying the design principle (a decision type) of distributing the AI model among the vehicles, using the cars for privacy-sensitive calculations. This decision implies other subsequent decisions, for instance the need of incorporating technologies over a Blockchain infrastructure to keep track of updates in data through a logging software component (supporting accountability then). As it stands out from the study, the solution given works for a particular context, namely highly distributed systems, from which the smart city design (in relation to smart vehicles) is an exemplification in the paper.

Example 2. Yokohama [25] addresses the problem of ensuring stability of the system when errors occur, in the context of AI-based systems organised according to a three-layered architecture.

What we learnt. The solution is based on a simple design principle, namely keeping separated AI-components from non-AI-components. Compared to the previous case, the solution is close to the design level and assumes one particular way to structure the overall architecture of the system (three layers). This also makes it possible to get a domain-independent solution. Later in the paper, they elaborate their design principle into a concrete architecture pattern, an AI-aware 3-layer architecture pattern.

Table 2. Ontology in some of the selected papers for our study.

Ref.	Architectural Decision	Decision Type	Context	Quality Attribute	Impact	Architectural Element	Architectural Layer
[16]	Distribute model	Design principle	Highly distributed systems	Privacy, Resilience	Supports	Vehicle	Physical
	Log updates	New technology		Accountability		Blockchain	Component
	Manage access			Scalability			
[25]	Keep AI and non-AI components separated	Design principle	N/A	Stability	Supports	Three layers	Logical
	3-layer AI-aware pattern	Architectural pattern				Subsystem	
[5]	Component replacement	Architectural tactic	Self-adaptation	Modularity	Supports	ML component	Component
	Retrain			Maintainability			Logical

Example 3. Casimiro et al. developed a preliminary framework aimed to self-adapt systems that rely on AI components [5].

What we learnt. They offer five adaptation tactics for AI-based systems. We discuss two of them. First, the "component replacement" tactic, consisting of replacing an under-performing component by one that better matches the current environment (dealing with concept drift). While this is fast and inexpensive, it may not be available in all scenarios. Second, the "retrain" tactic, which uses new data for retraining and updating the machine learning model's hyper-parameters. This is a generic and robust method, but effective only once a relatively large number of instances of the new data are available, computationally intensive, and with a significant increase of the accuracy and latency of the retrain process.

5 Discussion and research agenda

The work reported in this paper is answering two research questions. With respect to RQ1, this preliminary literature review has uncovered a number of remarkable facts:

- The types of architectural decisions are diverse and at different levels of abstraction and detail. In Table 2, we have provided some examples, but there are more, e.g. design pattern or architectural style. Eliciting and categorising these types is utterly important for our research goal.
- Similarly, knowledge about the different contexts and architectural levels in which architectural decisions are made need to be further elicited and consolidated. Concerning the context, we expect research papers to include a proper reflection on the limitations of applicability of their findings, through an appropriate statement of external validity threats.
- While the relevance of quality attributes as decisional drives has become evident in our literature study, papers usually focus on one particular attribute supported

by their approach, but they only occasionally discuss negative impact on other attributes. Quality trade-off analysis is well-known to be crucial in architectural decision-making [6], therefore we can expect research in this direction once the area becomes more mature.
- The papers that have surveyed focus on the static view of architecture decisions but they do not include much information about dynamic (process, using Kruchten's 4+1 model [15]) view. We also expect this aspect to be targeted in future works because it is of uttermost importance when deciding the most appropriate architecture for the system at hand.

On its turn, some points related to the ontology are worth to mention:
- At a first glance, the ontology seems to incorporate very little AI-related aspects. In fact, looking at the class diagram currently used for this preliminary proposal, only a subclass reflects our focus on AI-based systems. The reason is that, currently, AI-related concepts emerge in the instance level of the class diagram, as we can see in Table 2.
- Given the need to consolidate the current knowledge, as stated above in relation to RQ1, we have chosen not to predefine the values of the different classes e.g. using enumerate values (as we have done with the Impact type, which is the only exception to this rule since it is not really related to the AI domain but to the architectural decision domain only).

From this discussion, we highlight a few points that characterise our research agenda:
- Widen our literature review. The systematic mapping that we have used as baseline includes papers only until March 2020. Given the ever-growing plethora of research contributions in the AI field, we can expect a good number of papers that we have not considered yet.
- Complement the literature review with more practice-oriented knowledge sources, in the form of grey literature and interviews with practitioners.
- Consolidate the architectural knowledge from the literature review. As commented above, we would like to complete a catalogue of decision types, contexts, quality attributes and architectural views which gather all the knowledge related to architecting AI-based systems.
- Refine the ontology to include more specific and low-level AI concepts. This means reflecting in the ontology the consolidation mentioned in the point above. So, for example, we could specialise the concept of AI-related Architectural Element including e.g. Data Ingestion or ML model as subtypes.

6 Conclusions

This exploratory paper presents a summary of concepts related to architectural decisions in AI-based systems, and articulates them in the form of an ontology. This proposed, preliminary ontology can help to improve knowledge transfer among projects by harmonising concepts and actions used in diverse experiences, thus supporting *(i)* better understanding of the effects and implications of design decisions in different contexts; *(ii)* consolidation of architectural knowledge in specific domains and the subsequent definition of useful architectural and design patterns for AI-based systems.

We plan to apply to our research agenda to further deepen our understanding and conceptualization of the AI-based systems architectural decision-making area.

Acknowledgment

This paper has been funded by the Spanish Ministerio de Ciencia e Innovación under project / funding scheme PID2020-117191RB-I00 / AEI/10.13039/501100011033.

References

1. Ameller, D., Galster, M., Avgeriou, P., Franch, X. A Survey on Quality Attributes in Service-based Systems. *Software Quality Journal*, 24(2): 271–299 (2016).
2. Anthes, G. Artificial Intelligence Poised to Ride a New Wave. *Communications of the ACM*, 60(7): 19–21 (2017).
3. Bhat, M., Shumaiev, K., Hohenstein, U., Biesdorf, A. Matthes, F. The Evolution of Architectural Decision Making as a Key Focus Area of Software Architecture Research: A Semi-Systematic Literature Study. In: *Proceedings of the IEEE International Conference on Software Architecture* (ICSA): 69-80 (2020).
4. de Boer, R., *et al.* Architectural Knowledge: Getting to the Core. In: *Proceedings of the International Conference on the Quality of Software Architectures* (QoSA): 197–214 (2007).
5. Casimiro, M., *et al.* Self-Adaptation for Machine Learning Based Systems. In: *Proceedings of the 1st International Workshop on Software Architecture and Machine Learning* (*SAML*): paper 6 (2021).
6. Clements, P., Kazman, R., Klein, M. *Evaluating Software Architectures: Methods and Case Studies*. Addison-Wesley (2001).
7. Creus R., Martínez-Fernández, S., Franch, X.: Which Design Decisions in AI-enabled Mobile Applications Contribute to Greener AI? CoRR abs/2109.15284 (2021).
8. Franch, X., López, L., Cares, C., Colomer, D. The *i** Framework for Goal-Oriented Modeling. Book chapter in: *Domain-Specific Conceptual Modeling*, Springer (2016).
9. Dey, A.K., Abowd, G.D. Towards a better understanding of Context and Context-awareness. In: *Proceedings of the ACM Workshop on the What, Who, Where, When and How of Context-Awareness* (2000).
10. Di Noia, T., et al. A Fuzzy Ontology-based Approach for Tool-supported Decision Making in Architectural Design. *Knowledge Information Systems*, 58: 83–112 (2019).
11. ISO/IEC 25010:2011. Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models (2011).
12. ISO/IEC 42010: Systems and Software engineering - Recommended Practice for Architectural Description of Software-intensive Systems (2007).
13. Jurisica, I., Mylopoulos, J., Yu, E. Ontologies for Knowledge Management: An Information Systems Perspective. *Knowledge Information Systems*, 6: 380–401 (2004).
14. Kogut, P., *et al.* UML for Ontology Development. *The Knowledge Engineering Review*, 17(1): 61-64 (2002).
15. Kruchten, P.B. The 4+ 1 View Model of Architecture. *IEEE Software*, 12(6): 42-50 (1995).
16. Kumar, A., Braud, T., Tarkoma, S., Hui, P.: Trustworthy AI in the Age of Pervasive Computing and Big Data. In: *IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*: 1-6 (2020).
17. Martínez-Fernández, S., *et al.* Software Engineering for AI-based Systems: A Survey. *ACM Transactions on Software Methodology*, 31(2): 1-59 (2022).

18. Pahl, C., Giesecke, S., Hasselbring, W. Ontology-based Modelling of Architectural Styles. *Information and Software Technology*, 51: 1739–1749 (2009).
19. Schwartz, R., Dodge, J., Smith, N.A., Etzioni, O. Green AI. *Communications of the ACM*, 63(12): 54-63 (2020).
20. Serban V., Visser J.: Adapting Software Architectures to Machine Learning Challenges. In: *Proceedings of the 29th IEEE International Conference on Software Analysis, Evolution and Reengineering* (*SANER*) (2022)
21. Szyperski, C., Gruntz, D., Murer, S. *Component Software: Beyond Object-Oriented Programming*. Pearson Education (2002).
22. Tyree, J., Akerman, A. Architecture Decisions: Demystifying Architecture. *IEEE Software*, 22(2): 19–27 (2005).
23. Wan, Z., Xia, X., Lo, D., & Murphy, G.C. How does machine learning change software development practices?. *IEEE Transactions on Software Engineering*, 47(9), 1857-1871 (2019).
24. Warnett, S.J., Zdun, U. Architectural Design Decisions for the Machine Learning Workflow. *IEEE Computer*, *55*(3): 40-51 (2022).
25. Yokoyama, H.: Machine Learning System Architectural Pattern for Improving Operational Stability. In: *Proceedings of the IEEE International Conference on Software Architecture – Companion volume* (ICSA-C): 267–274.
26. Zdun, U., Stocker, M., Zimmermann, O., Pautasso, C., Lübke, D. Guiding Architectural Decision Making on Quality Aspects in Microservice APIs. In: *Proceedings of the International Conference on Service-Oriented Computing* (*ICSOC*): 73-88 (2018).