Adapting Unified Architecture Framework to SysML v2: A Project Financial System Case Study

Laurita Radiševska

Vilnius University, Faculty of Mathematics and Informatics,
Didlaukio g. 47, Vilnius
laurita.radisevska@mif.stud.vu.lt

Abstract. This paper explores how a SysML v2-based implementation of the Unified Architecture Framework (UAF) can be integrated into a project financial management system as a dynamic architectural component. A reusable SysML v2 library implementing UAF concepts has been developed from OMG specifications. The library is then used to create a domain model describing participating systems, data entities, organizational goals, project structures, and the budget lifecycle across four UAF viewpoints: Resources, Strategic, Projects, and Operational. A proof-of-concept web application is developed to integrate the model as a shared and continuously maintained source of architectural information. The results show that structural aspects of the model can be effectively reused within the system, while behavioral elements require additional implementation decisions and cannot be fully modified within the application.

Keywords: Unified Architecture Framework, SysML v2, architecture framework, model-based systems engineering, project financial system.

1 Introduction

Financial data management systems play a critical role in ensuring the completeness, timeliness, and consistency of project-related information. However, the software built to support it is often developed through incremental structural decisions driven by immediate needs, rather than through systematic upfront analysis. While ad hoc development may offer short-term speed, it becomes problematic as the number of participating systems grows – decisions are implicit, undocumented, and difficult to trace or verify [1]. Enterprise architecture systems are proposed as a solution to define a logical structure to the inefficiencies of ad hoc development by providing a logical construct for defining and controlling interfaces and

DOI: https://doi.org/10.15388/LMITT.2026.24

integrating all system components [2]. The Zachman Framework and The Open Group Architecture Framework (TOGAF) are the two most established and recognized enterprise architecture frameworks in use today [3]. The Zachman Framework is an ontology that organizes what needs to be described in an enterprise by mapping six viewpoints against six basic fundamental interrogatives, ensuring comprehensive coverage without prescribing a process for creating them [4]. TOGAF, by contrast, is a process-driven methodology whose Architecture Development Method (ADM) provides an iterative, step-by-step cycle for developing, implementing, and governing enterprise architecture [5]. The two are complementary: TOGAF provides the process for creating the architecture, while Zachman provides the scheme for organizing its artifacts and verifying completeness [5]. While both frameworks focus on enterprise structure and architectural management, they often lack the technical detail needed for modern digital engineering and for complex systems-of-systems integration. The Zachman Framework organizes what should be described, but does not produce connected, machine-readable models. TOGAF guides the development process but operates at a conceptual level that may not capture the precision required for actual system configurations. The Unified Architecture Framework (UAF) addresses these limitations by building on the Systems Modeling Language (SysML), which is a modeling language developed for systems engineering that supports the specification, analysis, design, verification, and validation of a broad range of systems and systems-of-systems. UAF uses SysML as its foundation and, like Zachman, organizes its models through a grid structure of domains and model kinds. However, unlike Zachman, its elements are formally linked across the grid, enabling digital traceability between strategic goals, operational activities, and physical resources [6]. This paper investigates how a SysML v2-based UAF model can be integrated into a project financial management system as an active architectural element. Rather than generating the system directly from the model, the approach treats the model as a structured source of organizational knowledge that informs system behavior, data structures, and access control. To evaluate this approach, a proof-of-concept web application is developed, demonstrating how architectural elements can be interpreted and incorporated into application layers. The results provide insight into which aspects of the model can be directly leveraged in implementation and where additional design decisions are required.

2 Background

Integrating different systems can be a complex task, even when they use the same language and terminology, because the lack of standardized rules and guaranteed interoperability leads to interpretation errors. This challenge became apparent during the development of military missions, the analysis of which showed that architecture development faces not only technical but also management-level obstacles. There was a lack of consensus on the content and use of the architecture, requirements were constantly changing, different countries applied uncoordinated national standards, and the tools in use lacked the ability to exchange model data, which prevented any coherent system interoperability. The need to reach a technical agreement among national interests led to the creation of an architecture system to align diverse system architectures [7]. In September 2012, a NATO Architecture Capability Team meeting in Brussels addressed incompatibilities among different architecture frameworks, laying the foundation for the development of the Unified Architecture Framework (UAF) [8]. This was followed by an intensive development phase, including a 2014 call for proposals for a new profile specification that would integrate multiple national frameworks. During this process, the Unified Profile for DoDAF and MODAF (UPDM) was renamed UAF, marking the transition to a universal architecture system. The first formal specification was released in October 2017 and continues to be refined by the Object Management Group (OMG) [9]. The work process was carried out using a model-driven approach and consisted of several stages, during which a new system comprising two complementary parts was developed based on the specification of various military architecture systems. First, to create a domain metamodel, UML metaclasses were defined to represent concepts adopted from the source architecture systems. Concepts and viewpoints were aligned to create a common domain metamodel usable by all contributing systems, separating the abstract level from the presentation layer. In this way, the goal of creating a domain metamodel independent of specific tools or metalanguages was achieved. For the presentation layer (UAF profile), the most relevant systems engineering modeling language at the time, SysML v1, was selected [8]. The specification format is technical but not sufficiently intuitive for gaining a high-level overview of the system. For this reason, a more manageable format – a view specification grid – was devised based on examples of donor architecture frameworks (AF). It allows a visual representation of how different view specifications relate to viewpoints in the horizontal rows and aspects in the columns, which describe a given view specification [6]. Based on the UAF grid, 11 viewpoints are distinguished: architecture management, summary & overview, strategic, operational, service, staff, resources, security, projects, standards, and actual resources. Each perspective is associated with corresponding aspects: motivation, taxonomy, structure, relationships, processes, states, sequence, information, constraints, path, and traceability. In 2025, after many years of development, the formal SysML v2 specification was released [10]. This version differs from the first in that it is based not on UML, but on a specialized metamodel called KerML (Kernel Modeling Language), which has a more formal mathematical foundation. SysML v1 relied almost exclusively on graphical diagrams, while SysML v2 introduced a standardized textual syntax that allows system models to be written as code, facilitates version control with Git, and enables more effective collaboration with software engineering practices [11]. As of March 2026, no official UAF profile for SysML v2 has been published by the OMG. The existing UAF specification remains based on SysML v1, leaving the adaptation of UAF concepts to SysML v2 as an open task [12].

3 Related Work

The Unified Architecture Framework (UAF) has been used as a structured approach for modeling complex systems-of-systems through formally defined viewpoints and connections. As a standardized architecture framework developed by the OMG, UAF provides a domain metamodel that supports traceable links between different architectural concerns [6].

Several recent studies have explored the practical use of UAF in system-level analysis and decision support. Bankauskaite and Morkevicius [13] proposed an automated UAF-based trade study process for system-of-systems architecture, demonstrating that UAF models can support structured evaluation of architectural alternatives. Similarly, Carroll et al. [14] showed how UAF can be applied to large-scale global problems by modeling interconnected systems and stakeholders using a systems-of-systems perspective. These studies confirm that UAF can be effectively applied beyond documentation and can support complex architectural reasoning.

In addition, Hause and Kihlström [15] investigated the application of UAF in modeling enterprise software systems, emphasizing the integration of organizational structures, operational processes, and technical resources within a unified architectural description. This is particularly relevant for systems such as project financial management platforms, where coordination across multiple domains is required.

Recent developments in SysML v2 have created new opportunities for implementing architecture frameworks in a more formal and machine-readable way. Morkevicius and Krisciuniene [12] analyzed the applicability of SysML v2 for UAF implementation and found that it provides clearer semantics and allows for a reduction in the number of concepts used. However, practical adoption is still limited due to immature tooling and the need for stricter modeling practices. Their work demonstrates the feasibility of representing UAF concepts in SysML v2 but focuses primarily on modeling rather than integration into software systems.

Overall, existing research shows that UAF is widely applied in system-of-systems analysis and enterprise modeling [13]–[15], while SysML v2 offers a promising formal foundation for implementing such models. However, limited research addresses how UAF models can be integrated into software systems and used as dynamic, executable architectural artifacts. Unlike existing work, which primarily focuses on modeling and analysis, this paper addresses the integration of UAF models into a working software system as a continuously used architectural component.

4 Methodology

The case study addresses an organization in which project financial management depends on the coordination of organizational units, project structures, and supporting information systems. The goal is to develop a web-based project financial management system that integrates a SysML v2-based UAF model as a dynamic source of architectural information.

The work consists of three stages, each building on the output of the previous one.

In the first stage, the OMG specifications for UAF [6], [9], [16] and SysML v2 [17] are studied and used to create a SysML v2 library that implements UAF concepts from scratch. The UAF Domain Metamodel specification [6] defines the concepts, relationships, and viewpoint structure; therefore, the SysML v2 specification [17] defines the language constructs available for implementing them. The library translates UAF language concepts into SysML v2 type definitions – organized by UAF viewpoint. The resulting library is reusable: it implements UAF concepts in SysML v2. It can be applied to create any domain-specific model, not only the project financial system used in this case study.

In the second stage, the library is imported into a domain model that represents the case study organization. The model instantiates the library types for a concrete project, assigns data ownership to specific departments, defines cross-departmental relationships, and describes the budget lifecycle process. Table 1 presents the initial selection of the UAF viewpoint and aspects for the domain model.

Table 1. UAF viewpoints and aspects selected for the domain model.

UAF viewpoint	Aspects	Role in the Model
Resources (Rs)	Taxonomy, Structure, Information	Participating departments and their responsibilities
Strategic (St)	Motivation, Structure	Organizational goals and drivers.
Projects (Pj)	Taxonomy, Structure	Project definitions, milestones, and the timeline of capability delivery
Operational (Op)	Processes	Budget lifecycle across departmental boundaries

In the third stage, a web application is developed to validate the proposed approach. The application integrates the SysML v2-based UAF model as a central architectural component, allowing it to function as a dynamic source of organizational structure and system behavior. The system supports key functionalities, including project initiation, budget planning, expense tracking, and invoice management. In addition, users can access architecture viewpoints derived from the UAF model, enabling role-based views of processes. The application is implemented using a layered architecture, consisting of a user interface, API, business logic, and data persistence layers. The backend is developed using Python and FastAPI, while the frontend is implemented using React. A relational database is used to ensure data consistency and integrity. The SysML v2 model is integrated into the application as a central source of architectural information rather than being used to directly generate implementation artifacts. To enable this integration, the model is parsed, and its structure is extracted, including elements, attributes, and relationships. The model content is used to inform different parts of the application. The Resources viewpoint provides information about participating systems, documents, and their boundaries, which supports the identification of integration points. The Strategic viewpoint defines organizational goals and capabilities, providing context for interpreting financial processes. The Projects viewpoint structures the system around project lifecycles, while the Operational viewpoint describes the budget lifecycle and inter-departmental processes, guiding the design of application workflows. The purpose of this stage is to evaluate how the information defined in the UAF-based SysML v2 model can be interpreted and utilized within the application, and to identify which elements can be directly supported and which require additional implementation decisions.

5 Results

The first stage produced a reusable SysML v2 library that implements UAF concepts as type definitions organized according to UAF grid viewpoints. The library covers four UAF viewpoints – Resources, Strategic, Projects, and Operational – with their corresponding aspects. Each UAF concept is translated into a SysML v2 part definition, connection, or flow, depending on its role in the metamodel. The library is independent of the case study – it provides the building blocks that any domain model can instantiate for a specific organization or system.

The second stage produced a model describing the case study organization. Under the Resources viewpoint, the model captures participating departments, their supporting IT systems, and the data entities they manage. The Strategic viewpoint defines organizational goals and capabilities that provide context for financial processes. The Projects viewpoint structures the system around project phases, milestones, and delivery timelines, while the Operational viewpoint represents the budget lifecycle as a sequence of interrelated activities across organizational units.

The SysML v2 model was parsed and integrated into the application as a live architectural artifact rather than a static design representation. The system provides role-based access to the model, enabling users to view architecture viewpoints relevant to their responsibilities and organizational roles. A system administrator is granted extended privileges to modify the model, with all changes being validated and immediately reflected within the application. This ensures that the model remains a continuously updated and shared source of truth, supporting both system functionality and architectural transparency across all users.

The integration demonstrates that the architectural model can be used as a shared and continuously updated source of information within the system. Structural elements such as data entities, relationships, and project structures can be consistently represented and accessed by users according to their roles. Process definitions can also be parsed and presented within the system; however, their modification is limited to content-level changes. Structural changes to processes, such as adding new steps or modifying execution logic, are not supported and require external updates to the model.

6 Conclusion

This paper demonstrated that a UAF model implemented in SysML v2 can be integrated into a software system not only as a design artifact, but as a continuously used and maintained source of architectural information. By embedding the model into a web-based application and providing role-based access, the approach enables users to interact with architecture viewpoints relevant to their responsibilities, while allowing controlled modification through administrative roles. This transforms the model from static documentation into a shared and evolving component of the system.

The main contribution of this work is the demonstration that a SysML v2-based UAF model can support application development by explicitly capturing structural and organizational aspects of the system and making them accessible within the implementation. This reduces reliance on implicit design decisions and improves transparency and consistency between architecture and system behavior. At the same time, the results highlight an important limitation. While structural elements of the model can be effectively reused, behavioral aspects remain only partially supported. Although processes can be parsed and represented within the system, their structural modification, such as introducing new steps or altering execution logic, cannot be performed directly and requires external changes to the model. This indicates a clear boundary between architecture modeling and executable system behavior.

Future work could focus on extending support for behavioral modeling, including mechanisms for safely modifying process structures within the system, as well as exploring integration with additional UAF viewpoints such as Security and Standards. Further evaluation could also compare this model-integrated approach with traditional development methods in terms of maintainability and development effort.

References

[1] V. Kulkarni, “Overcoming Technical Debt in Financial Systems: A Solution Architect’s Perspective”.

[2] M. Lankhorst, *Enterprise Architecture at Work*. in The Enterprise Engineering Series. Berlin, Heidelberg: Springer Berlin Heidelberg, 2017. doi: 10.1007/978-3-662-53933-0.

[3] D. Öri, B. Molnár, and Z. Szabó, “Formal Modelling Approach of Enterprise Architecture - Hypergraph based Representation of Business Information Systems:,” in *Proceedings of the 20th International Conference on Enterprise Information Systems*, Funchal, Madeira, Portugal: SCITEPRESS - Science and Technology Publications, 2018, pp. 727–735. doi: 10.5220/0006804607270735.

[4] M. Bennett, “About the Zachman Framework,” Zachman International - FEAC Institute. Accessed: Mar. 22, 2026. [Online]. Available: https://zachman-feac.com/zachman/about-the-zachman-framework

[5] Sourabh-Hajela, “TOGAF Vs. Zachman: Which Enterprise Architecture Framework Should You Choose? | CIO Portal.” Accessed: Mar. 22, 2026. [Online]. Available: https://cioindex.com/reference/togaf-vs-zachman/

[6] “Unified Architecture Framework (UAF) Domain Metamodel.” Accessed: Mar. 20, 2026. [Online]. Available: https://www.omg.org/spec/UAF/1.3/Beta1/DMM/PDF

[7] M. Hause, A. Morkevicius, and G. Bleakley, “Transitioning UPDM to the UAF”.

[8] M. Hause, G. Bleakley, and A. Morkevicius, “Technology Update on the Unified Architecture Framework (UAF),” *INCOSE Int. Symp.*, vol. 26, no. 1, pp. 1145–1160, Jul. 2016, doi: 10.1002/j.2334-5837.2016.00217.x.

[9] “About the Unified Architecture Framework Specification Version 1.3 beta.” Accessed: Mar. 19, 2026. [Online]. Available: https://www.omg.org/spec/UAF

[10] “Object Management Group Approves Final Adoption of the SysML V2 Specification | Object Management Group.” Accessed: Mar. 20, 2026. [Online]. Available: https://www.omg.org/news/releases/pr2025/07-21-25.htm

[11] K. Jankevicius, “Advent of SysML v2 | Lesson 7 - Week 1 Recap,” Sensmetry. Accessed: Mar. 20, 2026. [Online]. Available: https://sensmetry.com/advent-of-sysml-v2-lesson-7-week-1-recap/

[12] A. Morkevicius and G. Krisciuniene, “Towards UAF Implementation in SysML V2,” *INCOSE Int. Symp.*, vol. 34, no. 1, pp. 2110–2123, Jul. 2024, doi: 10.1002/iis2.13258.

[13] J. Bankauskaite and A. Morkevicius, “Towards an Automated UAF-based Trade Study Process for System of Systems Architecture,” *INCOSE Int. Symp.*, vol. 30, no. 1, pp. 391–405, Jul. 2020, doi: 10.1002/j.2334-5837.2020.00729.x.

[14] K. Carroll, A. Lyle, R. Lewark, C. Medina, and A. Morkevičius, “SoS - Global Solutions to Global Problems Using UAF,” *INCOSE Int. Symp.*, vol. 34, no. 1, pp. 2400–2412, Jul. 2024, doi: 10.1002/iis2.13277.

[15] M. Hause and L. Kihlström, “Modeling Enterprise Software with UAF,” *INCOSE Int. Symp.*, vol. 34, no. 1, pp. 2452–2475, Jul. 2024, doi: 10.1002/iis2.13280.

[16] “Enterprise Architecture Guide for UAF.”

[17] “SysML Specification.” Accessed: Mar. 06, 2026. [Online]. Available: https://www.omg.org/spec/SysML/2.0/Language/PDF