A Survey of Selection Criteria for Analysis Composition Operators in the Context of Palladio

Bahareh Taghavi
bahareh.taghavi@kit.edu
Karlsruhe Institute of Technology

Sebastian Weber
sebastian.weber@fzi.de
FZI Forschungszentrum Informatik

Abstract

To gain a comprehensive understanding of the quality properties of a system and its interactions, we require first analyzing it. Model-Based Analysis is an approach in Model-Driven Engineering that uses models to systematically analyze a system for structure, behavior, or quality characteristics. Due to the complexity and interdependence of modern systems, individual analysis approaches need to be combined to meet a specific purpose and achieve complete analysis. This paper provides an overview of current analysis composition operators in the context of Palladio, along with examples of how they are used. The objective is to define some criteria that aid in the judicious selection of the most fitting operator for distinct scenarios. In order to help engineers in making better-informed decisions about analysis composition within the Palladio context and ultimately directing them toward ideal operators.

1 Introduction

As software systems have expanded in domain and grown in complexity, the demand for tools and techniques to analyze their properties has increased. Analysis techniques are employed to reason about the structure, behavior, and quality of systems using models [7]. Model-based analysis, in particular, is an effective strategy to use for more efficient and effective decision-making and to avoid expensive design faults. Different modeling and analysis techniques can be combined by composing individual analysis components for specific analysis purposes, requiring the use of composition operators.

Palladio [2] is an approach to model, simulate, and analyze software architectures, keeping a focus on performance, maintainability, reliability, and other quality properties. The Palladio Component Model (PCM) is a widely used domain-specific modeling language for software systems. PCM associated analysis techniques enable dynamic analysis of system behavior and evaluation of different scenarios. Palladio has been expanded to encompass different dimensions of quality properties. These extensions also involve the expansion of Palladio's reach into various domains beyond its original focus. This broadening of scope increases the model's applicability and effectiveness across a wide variety of software development scenarios, in which composing and decomposing analysis techniques across different domains is essential for integrating desired analyses.

In this paper, we present an overview of analysis composition operators and a number of application examples of analysis composition in existing research projects that use some of these analysis composition operators. This paper's contribution involves examining the reasons behind the use or lack of specific analysis composition operators in existing application examples. Drawing from our insights gained in this exploration, we derive criteria to guide the selection of analysis composition operators for future application examples.

The paper is organized as follows: It begins with an introduction to analysis composition operators in Section 2. Following that, we offer an overview of existing application examples of analysis composition operators in Section 3. Section 4 details our selection criteria for choosing analysis composition operators. The paper concludes by summarizing the findings and outlining potential future research paths.

2 Overview of Analysis Composition Operators

Composition operators for analysis can be broadly categorized into several types [6]:

Composition by result exchange involves integrating separate analysis techniques by exchanging the results of their computations or simulations. This operator is suitable when one analysis technique relies on the results of another as an input and both techniques operate independently. Since there will be no real-time interaction between the analysis techniques, it represents the simplest way of analysis composition.

Composition by transformation focuses on model transformations, which involve converting models from one representation to another. It is important to note that the use of this composition operator can only be achieved if the models of all analysis techniques can be integrated using a joint formalism. This operator uses a single formalism model as input for analysis, which is often represented by Petri nets or queuing networks for example.

Composition by co-simulation is an operator that enables the integration of multiple analysis techniques in order to collectively model and analyze complex systems. Analysis techniques are coordinated to interact and exchange data in a synchronized manner to analyze the behavior of a larger, interconnected system. The coordinator plays a central role in the co-simulation approach. Its primary responsibility is to orchestrate and manage the execution of analysis techniques involved in the co-simulation process.

Composition by extension involves extending existing analysis techniques by adding the features of another analysis technique. This operator enhances the capabilities of analysis technique components, resulting in the creation of a unified and integrated system. However, this operator is suitable in cases where all analysis techniques are based on the same modeling paradigm and formalism.

3 Application Examples

In this section, we will explore several application examples of analysis composition operators in the context of Palladio.

IntBIIS: IntBIIS [3] extends the PCM and the Palladio performance analysis by a business process simulator to predict the performance of business processes and assess their impact on software systems through the use of a modeling language and analysis technique. The approach exemplifies composition by extension operator and is suitable to employ due to the same modeling paradigm of PCM usage and business process models. This is a white-box composition because it allows for a more accurate prediction of performance by considering the intricate relationships and behaviors of the components instead of using isolated analysis techniques.

PCA: The Power Consumption Analyzer (PCA) [4] uses the result of the Palladio performance analysis as input to a power consumption analysis to predict energy efficiency of a software system. The performance analysis results are combined with a power consumption metamodel to perform model-based energy efficiency analysis at the architecture level. It uses the composition by result exchange operator between isolated analysis techniques by treating the system as a black-box. To calculate power consumption based on the Power State Model in each state, PCA needs the measurements in the Palladio Runtime Measurement Model.

OMPCM: PCM only provides limited capabilities to model and simulate network communication. This limitation impacts the accuracy of predicting the performance of network-intensive systems. OMPCM [1] addresses this issue by combining Palladio architecture-level software performance prediction and OMNeT++-based network-centric simulator. This is an example of composition by co-simulation, enabling smooth integration with the established Palladio tool set and employing a grey-box approach. A bridge module coordinates the translation of events between the OMPCM and the network simulators. Using this bridge to remotely deploy components via the network connection can have an overhead that affects the response time of the system.

Message-Driven Self-Adaptive Systems: The approach is used to model the behavior of queuing and message brokers within self-adaptive system architectures. A simulation interface connects a simulation of a component-based architecture with a messaging-based simulation to offer quality forecasts [5]. This is an example of composition by co-simulation, in which the simulation interface must manage the consistency and synchronization of simulations.

Multi-Level Hardware Simulation: Modeling and analysis of systems are usually carried out on a single level of abstraction. Depending on this level of abstraction, accuracy and simulation of the analysis form a trade-off. Using multi-level hardware simulation, the level of abstraction and, consequently, the trade-off decision can be varied during a simulation. The composition of the different hardware simulators on different levels of abstraction is achieved by result exchange operator. Co-simulation is also used to compose the multi-level hardware simulation with the usage and software simulation of Palladio [8].

Coupling of Architectural Analyses and Static Source Code Analyses: To verify the security properties of assumed architectural security, a framework is introduced that combines static architectural analysis with specification-driven source code analysis [8]. This scenario employs composition by result exchange and a black-box approach due to the fact that the outcome of security-related information in the implementation, derived from the source code analysis result and their integration, is utilized as an input into the architectural analysis.

4 Selection Criteria

Based on the insights gathered from the application examples described in the previous section, we derive the following criteria to guide the selection of composition operators for future application examples.

Type of analysis:

> Black-box approach: When there is a composition between black-box analyses, it means that the internal structure is encapsulated and the implementation details are hidden from the outside [6]. The criterion derives from practical examples (such as PCA), which focuses solely on explicit interfaces without considering internal behavior and returns only the results. If the black-box criterion is found in an application example, selecting the composition by result exchange operator is the optimal choice.

White-box approach: When there is a composition between white-box analyses, we have access to the internal details, components, and logic of the analyses during the composition process. In other words, at each level of depth, the internals of the composed individual models are available for analysis and modification [6]. The application example of IntBIIS aligns with the white-box approach, which extends PCM's usage specification by incorporating business process constructs. In the case where a white-box approach is a criterion, both the composition by extension and composition by transformation operators can be selected based on other relevant criteria.

Grey-box approach: When there is a composition between grey-box analyses, it combines the advantages of white-box and black-box approaches [6]. There is a degree of transparency and internal knowledge of models, while a level of abstraction and modularity remains. For instance, OMPCM serves as an illustration of this kind of composition through co-simulation. Therefore, if the grey-box approach is determined as a criterion, the selection of composition by co-simulation is considered appropriate.

Coordination overhead: Coordination is needed when different analysis techniques are interacting and exchanging information with each other. Different mechanisms exist for synchronizing components, which may add additional overhead to the system depending on their complexity. It can be inferred from instances such as OMPCM, where the presence of a coordinator can have an impact on the response time of the service. This implies that, in situations where the impact of overhead on system performance is important, the selection of the co-simulation operator should be avoided.

Modeling paradigm: Modeling paradigm refers to the foundation and methodology of the systems which can be used in analysis composition process. Various analysis methods might adopt distinct modeling approaches based on how system components interact and the specific attributes of the systems involved. It is derived from the example of IntBIIS, which involves composition by extension, while both analysis techniques must adhere to the same modeling paradigm. Consequently, when both analysis techniques use the same modeling paradigm, it is practical to select composition by extension operator.

Compatible interface: In analysis composition techniques, a compatible interface is essential for efficient communication, data exchange, and facilitating a combined system. Interfaces that are compatible allow multiple components, systems, or modules to work together and interact with one another seamlessly. In Message-Driven example, implementing an interface can facilitate the architectural representation of messaging and ensure meaningful interoperability. Hence, in scenarios where having a compatible interface is not feasible, the composition by co-simulation operator does not represent the optimal choice.

Applying specific criteria is vital for selecting a suitable analysis composition operator that aligns with the analysis objective, system attributes, and domain prerequisites. This boosts accuracy and ensures consistent results.

5 Conclusion

In this paper, we presented an overview of analysis composition operators and discussed about several previous examples of those in literature. In addition, we derived some criteria from the provided examples to serve as guidelines. These criteria can be employed to assist in the selection of suitable operators for upcoming endeavors. As part of our future work, we aim to extend these criteria and introduce a novel classification of the efforts undertaken in this field. Furthermore, we intend to extend existing mechanisms for analysis composition and develop new modular semantic-based analyses.

Acknowledgements

This work was funded by the DFG (German Research Foundation) – project number 499241390 (FeCoMASS) and "Kerninformatik am KIT (KiKIT)" funded by the Helmholtz Association (HGF).

References

[1] J. Henss, P. Merkle, and R. H. Reussner. "The OMPCM simulator for model-based software performance prediction". In: *Proceedings of the 6th International ICST Conference on Simulation Tools and Techniques*. 2013, pp. 354–357.

[2] R. H. Reussner et al. *Modeling and simulating software architectures: The Palladio approach*. MIT Press, 2016.

[3] R. Heinrich et al. "Integrating business process simulation and information system simulation for performance prediction". In: *Software & Systems Modeling* 16 (2017), pp. 257–277.

[4] C. Stier. "Adaptation-aware architecture modeling and analysis of energy efficiency for software systems". PhD thesis. KIT, 2018, 2019.

[5] L. Schmid. "Modeling and Simulation of Message-Driven Self-Adaptive Systems". MA thesis. KIT, 2020.

[6] R. Heinrich et al. *Composing Model-Based Analysis Tools*. Springer, 2021.

[7] R. Heinrich. *Architecture-based Evolution of Dependable Software-intensive Systems*. 2023.

[8] S. Koch et al. *A Collection of Scenarios for the Decomposition and Composition of Model-based Analyses*. Tech. rep. 2. KIT, 2023. 14 pp.