



**HAL**  
open science

## MoVAL: Towards a Multi-views/Multi-hierarchy Software Architecture

Ahmad Kheir, Hala Naja, Kifah Tout, Mourad Chabane Oussalah

► **To cite this version:**

Ahmad Kheir, Hala Naja, Kifah Tout, Mourad Chabane Oussalah. MoVAL: Towards a Multi-views/Multi-hierarchy Software Architecture. 2014. hal-01006224

**HAL Id: hal-01006224**

**<https://hal.science/hal-01006224v1>**

Submitted on 14 Jun 2014

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# MoVAL: Towards a Multi-views/Multi-hierarchy Software Architecture

Ahmad Kheir<sup>1,2</sup>, Hala Naja<sup>1</sup>, Kifah Tout<sup>1</sup>, and Mourad Oussalah<sup>2</sup>

<sup>1</sup>Lebanese University, EDST, AZM platform for research in biotechnology, Tripoli, Lebanon

<sup>2</sup>LINA Laboratory, University of Nantes, Nantes, France

[ahmad.elkheir@univ-nantes.fr](mailto:ahmad.elkheir@univ-nantes.fr), [hjazzar@ul.edu.lb](mailto:hjazzar@ul.edu.lb), [mourad.oussalah@univ-nantes.fr](mailto:mourad.oussalah@univ-nantes.fr)

**Abstract:** In order to manage software systems' life cycle and complexities, software development communities has recognized the needs to invest in software architectures. Despite numerous research and industrial works established in software engineering domain, there are always many significant limitations in current software architecture approaches, like the lack of relations definition among different architectural components on the one hand, and the lack of multi-hierarchy definition for each architectural component on the other hand. This paper presents MoVAL, a multi-views/multi-hierarchy software architecture which complies with the IEEE recommended practice for architectural description (ISO/IEC/IEEE standard 42010-2011).

**Index Terms**—Software architecture, Viewpoint, View, abstraction level

## I. INTRODUCTION

Software architecture is the organization of different software elements inside a software, usually called a system, independently of the used technology. An element could be a component in Component Based Software Engineering (CBSE), a service in Service Oriented Architecture (SOA), or a module in Object Oriented practices (OO).

Actually, this organization is an intangible concept that needs to be communicated with different stakeholders. Thus, each software architecture should be represented via an architectural description (AD) that provides an overall picture of the system and its architecture. This AD provides intent, vision, and insight about the system, clarifies its purpose, goals, structural models, processes, and technology options.

Despite the importance of software architecture and its architectural description in modern and complex systems development, this native concept remains not sufficient to resolve current software complexities. For this reason, a major track was considered in software engineering that is based on the multi-viewpoints (*i.e.* multi-views) concept, in order to reduce those complexities.

## II. MOTIVATION

Despite the important advantages conducted by the viewpoint based software architecture approaches in general, these approaches still suffer some limitations. The study and analysis presented in [1], presents those limitations, which were the basis from which the objectives of our approach were made clear:

1. Existing approaches failed to treat the complexity that resides inside a single view. Indeed, in a view models are tangled (*i.e.* not organized according to some kind of relationships). Clearly, we believe that models that make up a view should be organized into well-formed hierarchies according to different description levels.
2. Some existing approaches propose an architectural description process based on Views, but none of them clearly embed description levels definition as a step during the latter process.
3. The viewpoint based software architecture approaches haven't resolved the inconsistency problems between different views of the architecture, although they admitted the existence of some complementary and dependency aspects between those architectural views.

Actually, our approach aims to define hierarchical multi-views software architecture in order to reduce the complexities of the system by breaking down its architectural description to several views and define for each view a hierarchy of levels. Also it aims define formal consistency rules between different views and hierarchy levels of the architecture in order to ensure the consistency of the

resulting software architecture.

### III. MoVAL APPROACH

MoVAL [2] is a Model, View, and Abstraction Level based software architecture approach that is based on the construction of multi-views architectures, and the decomposition of each view to multiple abstraction levels of several types.

Actually, MoVAL extends the IEEE 42010 standard [3] and also complies with other important standards in software engineering, like the meta-object facilities (MOF) [4].

A view in MoVAL is a representation of the system considering a set of the development process' aspects, and some problems associated to a specific category of stakeholders or a group of categories of stakeholders. This view is defined in multiple levels of abstraction of different types: the achievement levels and description levels. In fact, an achievement level groups a set of artifacts defining an architecture at a specific phase of the architecture definition process, while the description level allows the architect to provide multiple descriptions having different granularity levels for each achievement level. Note that views and abstraction levels of a MoVAL architecture are linked together via formal architectural element called links. Those links were shaped formally in order to hold the relations that may exist between different architectural elements (views and abstraction levels) and to ensure the consistency of the architecture.

### IV. MoVAL ARCHITECTURE DEFINITION PROCESS (ADP)

In order to benefit from MoVAL approach, it was crucial to define a methodology that guides the architect while developing his architecture. For this reason, a MoVAL specific architecture definition process (ADP) that complies with the Unified Process (UP) [5] was developed.

As it complies with the UP, MoVAL process is an iterative process composed of four different phases: inception, elaboration, construction, and transition.

### ACKNOWLEDGMENT

Authors would like to thank all who have contributed to the achievement of this preliminary and on-going study, especially the Lebanese National Council for Scientific Research (LNCSR) and the Lebanese association for scientific research (LASER) for funding this research project.

### REFERENCES

- [1] A. Kheir, H. Naja, M. Oussalah, and K. Tout, "Overview of an Approach Describing Multi-Views/Multi-Abstraction Levels Software Architecture," presented at the 8th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE 2013), France, 2013.
- [2] A. Kheir, H. Naja, and M. Oussalah, "Hierarchical Multi-Views Software Architecture," presented at the The Eighth International Conference on Software Engineering Advances (ICSEA 2013), Italy, 2013.
- [3] L. Jen and Y. Lee, "Working Group. IEEE recommended practice for architectural description of software-intensive systems," in *IEEE Architecture*, 2000.
- [4] OMG, "OMG Meta Object Facility (MOF) Core Specification," in *OMG's industry-standard environment*, 2013.
- [5] I. Jacobson, G. Booch, and J. Rumbaugh, "The Unified Software Development Process," *CERN Document Server*, 1999. [Online]. Available: <http://cds.cern.ch/record/385285>. [Accessed: 29-Jan-2014].