Computers and Electrical Engineering 58 (2017) 113–125

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

A bigraphical multi-scale modeling methodology for system of systems☆

Amal Gassara[a,*], Ismael Bouassida Rodriguez[a,b], Mohamed Jmaiel[a,b], Khalil Drira[c,d]

[a] ReDCAD, University of Sfax, B.P. 1173, 3038 Sfax, Tunisia
[b] Digital Research Center of Sfax, B.P. 275, Sakiet Ezzit, 3021 Sfax, Tunisia
[c] CNRS, LAAS, 7 avenue du colonel Roche, F-31400 Toulouse, France
[d] Univ de Toulouse, LAAS, F-31400 Toulouse, France

ARTICLE INFO

Article history:
Received 30 January 2016
Revised 19 January 2017
Accepted 20 January 2017
Available online 1 February 2017

Keywords:
Software architectures
System of systems
Multi-scale modeling
Formal methods
Bigraphical reactive systems

ABSTRACT

In this paper, we present a multi-scale modeling methodology for software System of Systems (SoS) using the formal technique of bigraphical reactive system. This methodology provides a correct by design approach ensuring the correctness of the SoS architectures. A first scale is defined by the designer. Then, it is refined by successively adding lower scale details. The transition between scales is implemented following a rule-oriented refinement process. The executed rules respect the system constraints ensuring, in this way, the correctness of the obtained scale architectures. Moreover, we address the dynamic aspect of SoS by providing model-based rules of reconfiguration actions. We illustrate our approach with a Smart Buildings case study.

1. Introduction

The increasing integration of software, hardware and network has raised a new class of software systems, the so-called System of Systems (SoS) [1]. A SoS is composed of large scale integrated systems that are heterogeneous and independently operable on their own, but are networked together for a common goal. Systems of Systems are increasingly involved in different areas [2] like emergency coordination and crisis management, health care, smart cities, etc.

These systems are characterized by geographic distribution, operational and managerial independence of their elements, evolutionary development, and emergent behavior. These characteristics have therefore brought new challenges to software development including software architecture design. In fact, a software architecture represents the structure of a system which comprises software components, their externally visible properties, and the relationships among them [3]. Hence, by providing a high-level model of the system architecture, it allows to understand the system in a simple way and to reason about its key properties.

However, the complexity of SoSs makes their design more difficult especially in ensuring the correctness of their architectures. By correct, we mean that architectures respect a set of functional requirements like structural constraints related to

☆ Reviews processed and recommended for publication to the Editor-in-Chief by Guest Editor Dr. M. Senthil Kumar.
* Corresponding author.
E-mail address: amal.gassara@redcad.org (A. Gassara).

http://dx.doi.org/10.1016/j.compeleceng.2017.01.016
0045-7906/

the hierarchy of their constituents and the connections between them. Therefore, an existing open issue is how to facilitate the modeling of SoS architectures and how to represent them rigourously in order to guarantee their correctness.

Another major issue is to address the dynamic aspect of such systems by ensuring reconfigurations. A reconfiguration describes how an architecture can evolve by adding or removing not only components and connections inside a system but also the containing systems themselves considered as larger components in their own.

In recent years, some research activities like [4,5] addressed the description of SoS software architectures. But they provide informal description and can be extended with a solid formal foundation to ensure the correctness of the designed architectures. Other approaches [6–8] were proposed including formal models and techniques. However, these approaches do not provide a solution for facilitating the modeling of SoS architectures and mastering their complexity. They addressed the dynamic aspect of such systems by focusing only on correctness by evolution and not on correctness by design. Whereas, our approach ensures in addition the correctness of the initial architecture.

In this direction, the main goal of this work is to provide a rigourous solution for SoS modeling based on the formal technique of Bigraphical reactive systems (BRS) [9] with an inspiring vision from multi-scale modeling [5,10–12].

Therefore, we proposed B3MS, a Bigraphical multi-scale modeling methodology for System of Systems. This formal methodology aims, first, to help the designer to model correct SoS architectures. Instead of modeling the whole SoS and verifying it with respect to the defined constraints, we rather propose a correct by design approach using a multi-scale modeling. Actually, multi-scale modeling is a suitable solution since it is based on a refinement process. Following B3MS, a coarse grained scale is defined by the designer. Then, it is refined automatically by successively adding lower scale details until detail goals are satisfied. This refinement is ensured by applying specific rules that comply with the structural constraints of ths system ensuring, in this way, the correctness of the obtained scale architectures.

Moreover, we address the dynamic aspect of SoS by giving some reconfiguration meta-rules which help the designer to model reconfigurations. These meta-rules are modeled as BRS and encompass the essential SoS reconfigurations such as adding, removing and replacing system constituents. In fact, we envision that modeling SoS reconfigurations can benefit, on the one hand, from multi-scale modeling since there is a need to work sometimes at coarse-grained levels and other times at fine-grained levels. On the other hand, modeling these reconfigurations benefits from the expressive power and visual representation of BRS to describe architectures and reconfiguration. It benefits also from the ability of BRS to abstract the constituent systems via parameters. Hence, the difficulty of dealing with SoS reconfigurations can be reduced by focusing on coarse-grained level without going into details of fine-grained levels.

The remainder of this paper is organized as follows. We present in Section 2 some research activities dealing with modeling SoS architectures. Then, in Section 3, we present B3MS, our proposed modeling methodology for SoS that is illustrated by the Smart Buildings case study in Section 4. Finally, Section 5 concludes this paper and presents future work.

2. Related work

Recent systematic literature reviews [13,14] on the description of SoS architectures presented the use of some architecture description languages (ADLs) such as UML (Unified modeling language), SysML (System modeling language), CML (COMPASS modeling language), and X-UNITY. So, we identified some research activities that are based on these ADLs. For instance, Khlif et al. [5] proposed a multi-scale modeling approach for SoS architecture description using UML and SysML notations. They ensure the description of SoS architectures (their constituents and the connections between them). To do so, they adopt a rule-oriented description technique to manage the refinement process as a model transformation between the coarse-grain and the fine-grain descriptions. Moreover, Bryans et al. [4] used SysML to specify interfaces among the constituents and then they enriched such a specification with CML to specify contracts in SoS software. CML [15] is specifically designed for SoS modeling and analysis. It is a collection of process definitions and interact with the environment via synchronous communications. The use of interface descriptions enables the specification of pre and post conditions for operations.

However, we can notice that these two studies support only the static aspect of SoS and do not address dynamic aspect. This is explained by the fact that ADLs lack evolution feature of SoS, as it is claimed by Guessi et al. [16] after assessing fours ADLs used for SoS architecture description. Actually, SoS architectures should support dynamic evolution by adding new constituents and connections or removing existing ones. Moreover, despite the advantages of semi-formal representation, mainly with regard to comprehension, formal methods are interesting since considering that many SoS can address critical domains.

In this direction, Flavio Oquendo and Axel Legay [17] defined SosADL, a novel formal ADL which is an evolution of Π-ADL, used to describe static and dynamic architectural specifications. It extends Π-ADL with new architectural concepts and notations for SoS. SosADL allows to declare a set of architectural constraints which are solved at runtime in order to construct architectures. Using SosADL, Petitdemange et al. [6] proposed an approach to describe SoS architectures and to manage their reconfigurations using the example of a fire emergency system. They proposed a pattern-based approach that provides a set of reconfiguration patterns in order to assist dynamic reconfigurations. According to these patterns, reconfigurations can be implemented and applied at runtime, while maintaining the consistency in the SoS.

Based on formal models, we identified also two major European projects (COMPASS [7] and DANSE [8]). DANSE [8] proposed a methodology for the development of SoS architectures based on architectural patterns. This methodology starts with an initial SoS model that is created using the Model SoS solution method. Then, this model is changed to encompass a pattern that offers improvement. Through simulation and statistical model checking, the changed SoS model is tested to