Computers and Electrical Engineering 000 (2017) 1–12

Contents lists available at ScienceDirect

Computers and Electrical Engineering

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

Describing correct deployment architectures based on a bigraphical multi-scale modeling approach☆

Amal Gassara[a,*], Ismael Bouassida Rodriguez[a,b]

[a] ReDCAD laboratory, University of Sfax, B.P. 1173, Sfax 3038, Tunisia
[b] Digital Research Center of Sfax, B.P. 275, Sakiet Ezzit, Sfax 3021, Tunisia

ARTICLE INFO

Article history:
Received 31 August 2016
Revised 12 May 2017
Accepted 12 May 2017
Available online xxx

Keywords:
Deployment architectures
Formal methods
Bigraphs
Multi-scale modeling
Correct by design

ABSTRACT

In this paper, we propose a formal approach supporting the correct description of deployment architectures and their reconfigurations. According to defined models, correct deployment architectures are generated and one of them is selected to be deployed. This generation process is based on a multi-scale modeling approach adopting Bigraphs and Bigraphical Reactive Systems as a modeling language. In fact, the architecture of a scale is refined by adding the components of the next scale. Then, the obtained architecture is in turn refined and so on, until reaching the last scale. The transition between scales is performed through applying refinement rules. Based on correct by design, the refinement process is executed on a correct scale architecture (respects the defined models) by applying correct rules. So, we ensure that the generated scale architectures are correct. Finally, our approach is illustrated through the case study Smart Home.

1. Introduction

With the increasing advances in technologies, software applications become more complex and distributed over a large network. This growth in size and complexity has brought new challenges to software deployment. Many studies considered that using the principles of software architectures is a promising approach to resolve these challenges [1]. A software architecture represents the structure of a system by a high-level model allowing to understand it in a simple way [2]. So, the architectural deployment provides a basis for understanding and analysing the physical distribution of the system through a set of hosts and devices, including the distribution of software components. A deployment architecture defines the mapping of software components into hardware components and the satisfaction of defined constraints [3].

However, with the continuous growth in the size and the complexity of software systems, describing correct deployment architectures is considered a challenging task. In fact, a correct deployment architecture must satisfy a set of constraints related to the software architecture like structural constraints (i.e, hierarchy of components and their connectivity) and to the target environment such as structural constraints and resource constraints as well as deployment constraints that concern the distribution of software components on hardware nodes.

☆ Reviews processed and recommended for publication to the Editor-in-Chief by Guest Editor Dr. S. Kallel.

* Corresponding author.
E-mail address: amal.gassara@redcad.org (A. Gassara).

http://dx.doi.org/10.1016/j.compeleceng.2017.05.018

In the literature, we note that there is a lack of methods that address the deployment issue from an architectural level. Most of existing approaches [3–7] focused on verifying deployment and resource constraints and few approaches [5,8] focused on verifying the structural constraints especially those related to the target environment architecture.

Furthermore, addressing the dynamic aspect of deployment architectures is considered a major issue. This aspect encompasses the runtime changes of deployment architectures by adding and removing components and changing their structural topology. The consideration of these reconfigurations raises the issue of their impact on the correctness of deployment architectures. However, most existing approaches [3,5–7] dealt with placement reconfigurations and did not address architectural reconfigurations and their correctness. Therefore, there is a need to propose a solution that facilitates the description of such reconfigurations ensuring the correctness of the evolved deployment architectures.

In this context, formal methods are all their interest to describe software architectures in a rigorous way and they have a potential impact on the quality of the system in an early phase of the software lifecycle. For this reason, we propose in our work a formal approach for describing correct deployment architectures and their dynamics.

In order to describe correct deployment architectures, our approach allows, at first, to define deployment models that describe the component types and their structural constraints. Then, according to these models, correct deployment architectures are generated and one of them is selected to be effectively deployed. This generation process is based on our proposed methodology in previous work [9] for describing multi-scale architectures. This methodology relies on a multi-scale modeling approach adopting Bigraphs and Bigraphical Reactive Systems (BRSs) as a modeling language. In fact, the architecture of the first scale is defined by the designer and it is refined by adding the components of the next scale. Then, the obtained architecture is in turn refined and so on, until reaching the last scale. The transition between scales is performed through applying refinement rules.

In order to ensure the correctness of generated architectures, our approach is based on a correct by design approach. In fact, the refinement process is executed on a correct scale architecture (respects the defined deployment models) by applying correct rules. So, we ensure that the generated architectures are correct.

Moreover, our approach tackles the dynamics of deployment architectures by proposing a set of reconfigurations that address communication changes. Based on a correct by design approach, we ensure that these reconfigurations preserve the structural constraints of the defined models in order to ensure the evolved architectures preserve these constraints, too.

The remainder of this paper is organized as follows. In Section 2, we present some research studies that deal with software deployment. We give, in Section 3, an overview of BRSs by introducing bigraphs and reaction rules. Then, in Section 4, we present our proposed approach for describing correct deployment architectures and we present in Section 5 our proposal for describing their dynamic reconfigurations. After that, our contributions are illustrated, in Section 6, by the Smart Home case study. Finally, Section 7 presents a conclusion and future work.

2. Related work

Many research activities addressed the issues of software deployment. We identified those using the Object Management Group Deployment and Configuration (OMG D&C) specification [10], those using Architecture Description Languages (ADLs) [2] and those using Domain Specific Languages (DSLs) [5,6].

2.1. Deployment using the OMG D&C specification

The OMG D&C specification is more suitable for work that focus on the execution of deployment and pay less attention for verification, as argued by Hnetynka in [11]. This specification does not allow to describe deployment concerns like resource and structural constraints neither the hierarchical structure of software components. Furthermore, it does not allow to model the dynamic aspect of applications. Therefore, most of the frameworks built upon this specification (like DAnCE [4,12] and Deployment Factory (DF) [11]) address static systems.

DAnCE is a middleware for Deploying and Configuring CORBA Components that enforces QoS requirements. Whereas DF is a model-driven unified environment that supports the deployment of software applications with different component technologies. However, we identified the framework DACAR [13] that dealt with the dynamic aspect of software systems. To do so, it used the concept of Event-Condition-Action (ECA) rules and a control loop.

2.2. Deployment using ADLs

Some research activities proposed architecture-based approaches [3,7,14] that rely on the concepts of software architecture to tackle the deployment issues. The work presented by Hoareau et al. [7] proposed a middleware for the deployment of software applications described by hierarchical components. This middleware is based on an extension of ADL to support the specification of context-aware deployments, i.e. deployments that are driven by resource constraints of the hardware hosts. Malek et al. [3] presented DIF (Deployment Improvement Framework) that allows to model deployment architectures through a formal model. Also, it proposes a set of algorithms to assess these architectures and find the optimal one w.r.t. multiple QoS dimensions.

Moreover, Benlahrache et al. [14] addressed the formalisation of software deployment described by AADL (Architecture Analysis and Design Language). They proposed a formal framework using BRSs (Bigraphs that are associated with reac-