Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 109C (2017) 400–407

Procedia Computer Science

www.elsevier.com/locate/procedia

The 8th International Conference on Ambient Systems, Networks and Technologies (ANT 2017)

Model consistency for multi-scale architectures applied to smart systems

Ilhem KHLIF[a,b,c,*], Mohamed HADJ KACEM[a], Marwa KALLEL[a], Khalil DRIRA[b,c], Ahmed HADJ KACEM[a]

[a] *University of Sfax, ReDCAD Research Laboratory, Sfax, Tunisia*
[b] *CNRS, LAAS, 7 avenue du colonel Roche, F-31400 Toulouse, France*
[c] *Univ de Toulouse, LAAS, F-31400 Toulouse, France*

Abstract

We propose a multi-scale methodology for complex software systems using the model transformation techniques. The methodology provides a correct by design approach for the description of multi-scale architectures. It is founded on a rule-oriented iterative modeling process ensuring transition from a coarse-grained description to a fine-grained description. The rules manage the refinement between scales and are applied with respect to the system constraints. Multi-scale architectures are modeled graphically using UML notations. The design process supports model transformation and validation of UML models with the OCL constraints defined on UML models. We illustrate our approach with a case study dedicated to the smart cities.

Keywords: Software Architecture, multi-scale description, UML models, OCL constraints , model transformation, refinement.

1. Introduction

Software systems are increasingly become complex and distributed on different levels. Complex systems are described in a recursive way as an interconnection of subsystems by means of architectural elements. The complexity of these systems makes their design more difficult especially in ensuring the correctness of their architectures. The software architecture can characterize the system design at a high level, and focus on the components as well as their interactions. Architecture descriptions provide the conceptual abstraction for modeling complex software systems. We focused our research study on model-based design of software systems. An iterative modeling process that helps architects to elaborate complex but yet tractable and appropriate architectural models can be implemented by successive refinements. Different properties of correctness and traceability have to be maintained between the models and the specifications at the different levels of iterations. Providing rules for formalizing and conducting such a process is our objective, which we implemented in visual modeling notations and formally specified in a formal way. For

* Corresponding author. Tel.: +216-27-854-465.
E-mail address: ikhlif@laas.fr

1877-0509 © 2017 The Authors. Published by Elsevier B.V.
Peer-review under responsibility of the Conference Program Chairs.
10.1016/j.procs.2017.05.408

this purpose, we propose to consider different architecture descriptions with different levels of modeling details called **"the scales"**. We define an iterative process starting from a coarse-grained description and leading to a fine-grained description. The proposed approach mainly consists of two steps. In the first step, multi-scale architectures are modeled graphically using UML notations. In the second step, the models are enriched with constraints specified with the Object Constraint Language (OCL). We experiment our approach with a case study dedicated to the smart cities. The remainder of the paper is organized as follows. We describe the methodology in Section 2. Section 3 presents the case study. In Section 4, we present a survey of related work. We conclude and outline some perspectives in Section 5.

2. Design process

We propose a multi-scale modeling approach for software architectures. The proposed design approach is founded on UML notations and uses component diagrams. The diagrams are submitted to vertical and horizontal transformations for refinement; this is done to reach a fine-grain description that contains necessary details. The model transformation ensures the correctness of UML description, and the correctness of the modeled system. UML provides a formal language, the Object Constraint Language (OCL), to define constraints on model elements. Our approach supports model transformation and validation of UML models with OCL constraints.

2.1. UML modeling

We present the multi-scale approach by a two-dimensional array describing vertical and horizontal scales[1]. The top-down scale transformation process, much like regular refinement, begins with a high level description of a system which we describe as a whole. Then, scale changes are applied to obtain a more detailed description, by describing components and connections. An iterative modeling allows to refine software systems descriptions. The vertical scales add the architecture decomposition details to obtain a more detail on the internal architecture of previously defined components. The horizontal scales describe or give details on the interconnections between components and their interfaces. We iterate on the architecture until reaching the details necessary to verify the associated architectural properties. The first scale Sv_0 begins with specifying the application requirements. It defines the whole application by its name. Two horizontal refinements called horizontal scales are associated with the first scale Sv_1. The first horizontal scale shows all components that compose the application. The second one describes the links between those components. Four horizontal refinements are associated with the second scale Sv_2. The first scale presents subcomponents for components, and enumerates all the roles that each component can take. The second one identifies the list of communication ports for each component, and refines those roles. The third one shows the list of interfaces for communication ports. The last one is obtained by successive refinements while adding the list of connections established between components and subcomponents. This scale allows us to define the architectural style. With a graphical modeling language like UML, we are not able to express all the information required to convey the exact information of the domain in the diagram. We propose to use OCL to define constraints on models to define the well-formedness rules of our iterative design process.

2.2. OCL constraints

To ensure the correctness of the modeled system, we adopt a rule-oriented description technique. The rules manage the refinement between scales. We use OCL to complement the information in UML models and we define model constraints in a form of invariants that each instance of the element in the context must satisfy. We express (classes, interfaces, associations) from an UML model as types in the OCL expressions written in the context of an instance. The invariants are the conditions that must be fulfilled at all times, and describe the structure of the system.

2.2.1. Vertical model refinement

Following our approach, a first scale is defined by the designer. Then, it is refined by successively adding design details until detail goals are satisfied. This refinement is performed by applying both vertical and horizontal model refinement rules that respect the constraints ensuring, in this way, the correctness of the obtained architecture. R_0 **rule:** The first scale Sv_0 is defined by the designer and represents the whole `Application` as a unique class. The first rule we consider is defined on the vertical scale 'Sv':