Information and Software Technology 75 (2016) 105–121

Contents lists available at ScienceDirect

Information and Software Technology

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

Run-time requirements verification for reconfigurable systems

George Chatzikonstantinou*, Kostas Kontogiannis

National Technical University of Athens, Department of Electrical and Computer Engineering, 9 Iroon Polytechniou, Athens 157 80, Greece

ARTICLE INFO

Article history:
Received 24 May 2015
Revised 6 March 2016
Accepted 10 April 2016
Available online 12 April 2016

Keywords:
Software engineering
System analysis
Run-time requirements verification
Goal models reasoning
Fuzzy reasoning

ABSTRACT

Context: Modern software systems often are distributed, run on virtualized platforms, implement complex tasks and operate on dynamically changing and unpredictable environments. Such systems need to be dynamically reconfigured or evolve in order to continue to meet their functional and non-functional requirements, as load and computation need to change. Such reconfiguration and/or evolution actions may cause other requirements to fail.

Objective: Given models that describe with a degree of confidence the requirements that should hold in a running software system, along with their inter-dependencies, our objective is to propose a framework that can process these models and estimate the degree requirements hold as the system is dynamically altered or adapted.

Method: We present an approach where requirements and their inter-dependencies are modeled using conditional goal models with weighted contributions. These models can be translated into fuzzy rules, and fuzzy reasoners can determine whether and to what degree, a requirement may be affected by a system change, or by actions related of other requirements.

Results: The proposed framework is evaluated for its performance and stability on goal models of varying size and complexity. The experimental results indicate that the approach is tractable even for large models and allows for dealing with models where contribution links are of varying importance or weight.

Conclusion: The use of conditional weighted goal models combined with fuzzy reasoners allowed for the tractable run-time evaluation of the degree by which system requirements are believed to hold, when such systems are dynamically altered or adapted. The approach aims to shed light towards the development of run-time requirements verification and validation techniques for adaptive systems or systems that undergo continuous, or frequent evolution.

1. Introduction

Over the past few years we experienced a significant growth on the deployment of highly interconnected systems operating in a number of domains such as banking, commerce, and government to name a few. These systems encompass complex requirements yielding thus large and complex models [1]. In addition to the complex requirements these systems entail, advances in virtualization technology make also possible the continuous evolution, dynamic provision, and dynamic adaptation of system resources, in a quest for these systems to constantly meet their numerous, and possibly diverse, functional and non-functional requirements. The problem has been recognized in the related research literature as an important one [2,3], both from the requirements verification perspective and, from the policy compliance perspective. Hence, it is important to develop techniques, and tools for assessing at run time whether changes in the system structure, and, operating environment violate requirements set forth by its stakeholders.

To date, the software engineering community has responded by investigating models, and frameworks allowing for the specification, analysis, and evaluation of system requirements. In this respect, research efforts have focused so far on the off-line static analysis and reasoning in such models for the purpose of evaluating the completeness, and validity of system requirements against stakeholder goals [4,5]. However, as business processes become more complex requiring highly inter-connected and inter-dependent services (e.g. Systems-of-Systems), issues related to the run-time verification of global system properties have emerged as very timely, important and challenging [6–8]. Work in the area

* Corresponding author. Tel.: +302107722511; Fax: +30 210 772 2511.
E-mail addresses: gchatzik@cslab.ece.ntua.gr (G. Chatzikonstantinou), kkontog@softlab.ntua.gr (K. Kontogiannis).

http://dx.doi.org/10.1016/j.infsof.2016.04.005
0950-5849/

of adaptive systems is attempting to address some of these issues, and research efforts aiming to model and analyze the run-time behavior of such systems have started emerging in the related literature [9,10]. However, there is still limited work on dealing with large requirements models, especially when the dependencies and impact of one design decision or system requirement cannot be fully and deterministically modeled due to the high complexity of structural and behavioral inter-dependencies present in such complex systems. It is therefore important to investigate analysis techniques that are tractable so that can be applied at run-time, and allow for reasoning on large requirements models, especially in the presence of incomplete knowledge related to the impact changes in one system or component may have on the behavior and requirements of other interconnected systems or components.

For this paper, we adopt a goal-driven view of this run-time verification problem aiming to assess whether a system that operates in a specific context is compliant to a set of requirements. This issue entails a number of challenges that have to be considered and addressed. First, an appropriate modeling notation for denoting goals that should hold in the running system under various contexts must be used. Such a modeling notation should be expressive enough to capture dependencies that exist between goals, allow for the definition of multiple variations of the model under different contexts, and allow for modeling the way design decisions and system properties affect these goals. Such a notation must also have well defined semantics so as to enable the use of tools and algorithms that can automate the processing of the models. Second, as dependencies in a complex system may not be fully known, or properly recorded, a fuzzy approach should be used for specifying such dependencies within an acceptable degree of ambiguity. Third, a reasoning framework is required to evaluate at run-time how and to what degree system changes affect the requirements taking into account events collected from the running system.

In a nutshell, the proposed framework allows for a) system goals and their variations under multiple contexts, hereinafter referred to as *views*, to be modeled as conditional goal trees [11], whose nodes hold with a degree of truth; b) dependencies between goals to be modeled as weighted contributions [12,13], where weights are interpreted as the degree by which a stakeholder's belief for the target node satisfaction/dissatisfaction is *increased* given the satisfaction status of the source node; c) transformations to be utilized in order to map conditional goal trees to weighted fuzzy rules [14] and; d) fuzzy reasoning to be applied in order to tractably verify whether, and to what degree, specific system goals hold at run-time.

The paper is organized as follows. Section 2 formally defines the requirements-based view of the verification problem. Section 3 summarizes key concepts in the areas of Goal Models and weighted fuzzy rules. Section 4 briefly describes the details of our approach, and Section 5 presents a running example used throughout the paper. Section 6 introduces the semantics of the notation utilized to model system goals and views, and Section 7 describes the transformation of goal models to rules that can be used by the proposed inference engine. Subsequently, a lab experiment is presented in Section 8, and the performance of the proposed framework is evaluated against randomly generated models of varying size and complexity in Section 9. Finally, Section 10 presents related work and Section 11 concludes the paper.

2. Formal definition of the problem

Run-time verification is the process of evaluating, while the system operates, whether it meets certain expected behavior and goals [15,16]. In this paper we adopt a requirements-based view of the run-time verification problem that we refer to as the *ReqRV* problem. To formally define it, we use as a starting point the seminal study by Zave and Jackson on software requirements [17].

According to this study, given a set of requirements R, and a set of domain assumptions D, the requirements satisfaction problem aims to determine the specifications S that can ensure the fulfilment of requirements R. The above can be summarized by the following equation as originally presented in [17]:

$$D \cup S \vdash R \tag{1}$$

In a similar manner as in [18], Eq. (1) can be adapted to the ReqRV problem, where now the problem can be formulated as follows. Given a set of domain assumptions D, and the description of the system in terms of a set S of observable characteristics of the system (e.g. logged events), deduce the values of the requirements in R (i.e. get the values that are logically implied by D and S for R), hereinafter referred to as *system goals*. However, in a software system, goals that hold at run-time and the dependencies that exist between these goals, may vary depending on the context C the system operates in, hence the problem can be formally defined as:

$$D(C) \cup S \models R(C) \tag{2}$$

where $D(C)$ are the rules that describe the knowledge related to those goals in context C, S is the values of the observable characteristics that reflect the state of the running system, and $R(C)$ the satisfaction values (i.e. true if goal is satisfied, false otherwise) for all system goals in context C. Note that in Eq. (2) S does not depend on context C, as we consider that the observable characteristics collected via the monitoring infrastructure of the system do not depend on context changes, and the same data are collected no matter what the context is. Subsequently, having deduced the values for all system goals from Eq. (2), we can check whether they are satisfied, i.e. their value in $R(C)$ is true.

However, in our analysis, we consider a *fuzzy* approach towards system goals' satisfaction, i.e. a system goal is not either satisfied (true) or denied (false), but rather it may be satisfied to a certain degree in the interval [0,1] expressed as a percentage, e.g. a goal may be 80% satisfied. Additionally, rules in $D(C)$ are annotated with a weight in the interval (0,1] which denote the subjective degree of belief a domain expert has in a rule. In this context, by applying Eq. (2), we get for each system goal in $R(C)$, a truth value in the interval [0,1] denoting its satisfaction degree in the running system. We can now check whether the system complies with the predefined set of requirements, by checking whether either the inferred degrees are greater that an specific threshold or if the satisfaction degrees are within an interval of acceptable values.

3. Background

3.1. AND/OR goal trees

AND/OR goal trees are a modeling formalism extensively utilized in requirements engineering, where its basic concept is the top-down AND/OR decomposition of goals into sub-goals. An AND-decomposed goal can be satisfied if all of its sub-goals are true, while an OR-decomposed goal is fulfilled if at least one of its sub-goals holds.

Additionally, two goals may be connected by a contribution link. More specifically, a goal may potentially contribute to other goals in four ways, namely, S^P, S^N, D^P and D^N. In this paper we adopt the semantics stated in [19] for those four contribution types, and annotate each contribution from a source node g_s to a target node g_t with a number called weight (w) like in [12,13],