Information and Software Technology 80 (2016) 245–264

Contents lists available at ScienceDirect

Information and Software Technology

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

GODA: A goal-oriented requirements engineering framework for runtime dependability analysis

Danilo Filgueira Mendonça[a,c,*], Genaína Nunes Rodrigues[a], Raian Ali[b], Vander Alves[a], Luciano Baresi[c]

[a] Department of Computer Science, University of Brasilia, Brazil
[b] Faculty of Science and Technology, Bournemouth University, United Kingdom
[c] Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Italy

ARTICLE INFO

Article history:
Received 7 February 2016
Revised 5 September 2016
Accepted 13 September 2016
Available online 17 September 2016

Keywords:
Goal modeling
Dependability
Probabilistic model checking
Runtime analysis

ABSTRACT

Context: Many modern software systems must deal with changes and uncertainty. Traditional dependability requirements engineering is not equipped for this since it assumes that the context in which a system operates be stable and deterministic, which often leads to failures and recurrent corrective maintenance. The Contextual Goal Model (CGM), a requirements model that proposes the idea of context-dependent goal fulfillment, mitigates the problem by relating alternative strategies for achieving goals to the space of context changes. Additionally, the Runtime Goal Model (RGM) adds behavioral constraints to the fulfillment of goals that may be checked against system execution traces.

Objective: This paper proposes GODA (Goal-Oriented Dependability Analysis) and its supporting framework as concrete means for reasoning about the dependability requirements of systems that operate in dynamic contexts.

Method: GODA blends the power of CGM, RGM and probabilistic model checking to provide a formal requirements specification and verification solution. At design time, it can help with design and implementation decisions; at runtime it helps the system self-adapt by analyzing the different alternatives and selecting the one with the highest probability for the system to be dependable. GODA is integrated into TAO4ME, a state-of-the-art tool for goal modeling and analysis.

Results: GODA has been evaluated against feasibility and scalability on Mobee: a real-life software system that allows people to share live and updated information about public transportation via mobile devices, and on larger goal models. GODA can verify, at runtime, up to two thousand leaf-tasks in less than 35ms, and requires less than 240 KB of memory.

Conclusion: Presented results show GODA's design-time and runtime verification capabilities, even under limited computational resources, and the scalability of the proposed solution.

1. Introduction

Many failures in software systems stem from poor requirements elicitation [15] and thus a proper understanding of what the system is supposed to do is key for its *dependability*. To this end, GORE (Goal-Oriented Requirements Engineering, [31]) offers proved means to decompose technical and non-technical requirements into well-defined entities (goals) and reason about the alternatives to meet them.

More recently, GORE has been used as means to model and reason about the systems' ability to adapt to changes in dynamic environments [1,36]. Goals have been used as both design and runtime artifacts. Goal modeling has been used to customize software systems with respect to the characteristics of the organization they are deployed in [3], to derive high-variability designs, and to maximize the resilience and adaptivity of deployed systems [46,49]. It has also been used as runtime model to respond to dynamic changes —while maintaining dependability. For example, goals become live entities that can self-adapt according to the context [7,8], or are complemented with meta-requirements that refer to their

* Corresponding author.
E-mail addresses: danilo.filgueira@polimi.it (D.F. Mendonça), genaina@unb.br (G. Nunes Rodrigues), r.ali@bu.ac.uk (R. Ali), valves@unb.br (V. Alves), luciano.baresi@polimi.it (L. Baresi).

http://dx.doi.org/10.1016/j.infsof.2016.09.005
0950-5849/

success or failure and can recover from errors [44]. The Runtime Goal Model (RGM) [14] augments goals and tasks with runtime specifications to verify whether their instances behave correctly, that is, they are dependable.

In previous work [36], we proposed the Dependability Contextual Goal Model, which exploits fuzzy logic to reason about the effects the actual context of operation has on both dependability requirements and dependability attributes. However, after studying some further real-life case studies, we have understood that the approach could become prohibitively heavy and time-consuming due to the effort required in providing declarative rules for each different goal, attribute, and context. Defined rules could also be corrupted by imprecise domain knowledge.

In addition, most of the solutions for eliciting dependability requirements do not take into account the history of failures. This is mandatory to be able to foresee probabilities of success and failing trends, and thus to support decision making procedures that can identify appropriate strategies to keep the system dependable. As a consequence, we advocate that dependability requirements models must be probabilistic, and that sound approaches and new tools be developed to guide self-adaptive capabilities and guarantee the fulfillment of goals.

In this context, *probabilistic model checking* (PMC) is suitable for reasoning about dependability requirements since it helps compute the probabilities with which these properties are satisfied [6]. PMC has been largely supported by tools such as PRISM [30] and PARAM [23]. The challenge is thus the conceptualization and formulation of dependability requirements in a way suitable for PMC.

This paper proposes the Goal-Oriented Dependability Analysis framework (GODA) to model goals and analyze their fulfillment in different contexts. GODA takes into account runtime aspects and accommodates the implications that contextual information may have on goal satisfaction. Since the overall goal satisfaction may be impacted by context restrictions, GODA provides a means to specify the interplay between them and to estimate the dependability of the strategies to fulfill goals in different contexts. At runtime, the outcome provided by GODA can be used to analyse whether the system is fulfilling its dependability goals. If it turns out that the obtained dependability is under a certain threshold, the system should consider the strategy (or strategies) that provide the most suitable dependability measure.

The proposed analysis solution relies on discrete-time PMC, where required specifications are obtained automatically from contextual runtime goal models. These goal models borrow concepts from contextual goal models [1] and runtime goal models [14]. Obtained models are verified through parametric PMC to take into account the possible variability of the probabilities in the model. However, since parametric PMC solutions are not fast enough and do not scale as needed, we propose an innovative solution for computing the parametric formulae. Our solution uses model checking to precompute the formulae that define the dependability of any node of the goal tree and takes into account the type of decomposition and runtime and context annotations. It then composes the different probabilities through suitable rewriting by mimicking the tree structure of the goal model. GODA is implemented as an extension to TAOM4E [39]: a TROPOS-based requirements elicitation and modeling tool implemented on top of the Eclipse framework.

In this paper, we also report on the empirical evaluation we carried out. First, we evaluate GODA on Mobee: a real-life software system that allows people to share live and updated information about public transportation via mobile devices. Mobee has been running for over a year and has already more than three thousand users. Our results in Mobee show that GODA is capable of performing dependability analysis efficiently, allowing it to be used under limited computational resources. The second part of the empirical study presents a time-space scalability analysis of the parametric verification. We artificially created goal models up to two thousand leaf-tasks, simulation results show a verification time below 35ms, and a use of less than 240 KB of memory in the worst-case.

The rest of the paper is organized as follows. Section 2 recalls the necessary background. Section 3 presents the conceptual model behind GODA, the proposed dependability analysis, and sketches our implementation as extension to TAOM4E. Section 4 describes the evaluation we carried out. Section 5 surveys related approaches and Section 6 concludes the paper.

2. Background

2.1. Goal modeling and context

Goal modeling provides a means to analyze the many requirements of the different stakeholders of a software system [10,15,48]. As defined by TROPOS [10], goals are owned by actors and actors may inter-depend on each other to reach their goals. Goals are ultimately fulfilled by *leaf-tasks*, which denote processes to be executed by actors. Goals and tasks are decomposed and organized in a tree-structured model. A goal can be decomposed in subgoals or refined by a *means-end* task. Means-end relationships link a goal to a (means-end) task whose fulfillment is a sufficient and necessary condition to the fulfillment of the goal. A non-leaf task can be decomposed into other subtasks. An *AND-decomposition* requires that all sub-nodes to be fulfilled, while an *OR-decomposition* requires that at least one sub-node be fulfilled. Accordingly to TROPOS, only one type of decomposition per node is allowed. The alternative paths (OR-decompositions) in the goal tree can be evaluated with respect to qualitative objectives called *soft-goals*. Soft-goals are goals without a clear-cut criteria for their fulfillment. *Contribution links* identify the positive or negative impact of alternatives on soft-goals.

Fig. 1 presents the different concepts related to goal modelling used by GODA. A single system actor (Mobee Mobile) has a root goal G_1, which is AND-decomposed into G_3 and G_4, meaning both subgoals must be achieved to fulfill G_1. Goal G_3 is AND-decomposed into G_8 and G_9, which are linked to tasks T_1 through means-end decomposition links. Other goals in the model are refined in a similar way. Tasks are also refined through AND/OR-decompositions, except leaf-tasks such as $T_{1.21}$ and $T_{1.22}$. Despite the support for multiple actors in TROPOS, in this work we consider a single system actor, i.e., all goals, tasks and relationships belonging to a system actor model.

An OR-decomposition represents alternative ways of fulfilling actor's goals [49]. For instance, task $T_{1.1}$: *Fetch geolocation* (a subtree of means-end task *T1: Track line location*) has two alternative tasks to track geo-location: *Fetch GPS* and *Fetch triangulation*. Each task contributes to the soft-goal *Geolocation accuracy* through positive or negative contribution links. Additionally, the fulfillment of a goal, the alternatives to do it, and the quality of each alternative can all be context-dependent [1]. In a Contextual Goal Model (CGM), context is defined as a formula of world predicates, henceforth defined as context facts. For example, if the context formed by fact *GPS Available* does not hold, the only way of fetching the geo-location would be through triangulation. Or, if *Battery* ≤ 15 holds, only manual information should be collected and goal G_4 is disabled, meaning that G_3 alone satisfies G_1 in that context.

2.2. Probabilistic model checking

Our proposal advocates that dependability requirements models be probabilistic, and that sound approaches and new tools be developed to guide self-adaptive capabilities and guarantee the fulfillment of goals. This is because goals are ultimately fulfilled by executable processes, which can fail due to different reasons. As