Data & Knowledge Engineering xxx (xxxx) xxx–xxx

Contents lists available at ScienceDirect

Data & Knowledge Engineering

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

Planning runtime software adaptation through pragmatic goal model

Felipe Pontes Guimarães[a,*], Genaína Nunes Rodrigues[b], Raian Ali[c], Daniel Macêdo Batista[a]

[a] Department of Computer Science, University of São Paulo (IME-USP), Brazil
[b] Department of Computer Science, University of Brasilia, Brazil
[c] Faculty of Science and Technology, Bournemouth University, United Kingdom

ARTICLE INFO

Keywords:
Requirements engineering
Adaptive systems
Context-awareness
Quality of service

ABSTRACT

Adaptivity is a capability that enables a system to choose amongst various alternatives to satisfy or maintain the satisfaction of certain requirements. The criteria of requirements satisfaction could be pragmatic and context-dependent. Contextual Goal Models (CGM) capture the power of context on banning or allowing certain alternatives to reach requirements (goals) and also deciding the quality of those alternatives with regards to certain quality measures (softgoals). It is used to depict facets of the decision making strategy and rationale of an adaptive system at the preliminary level of requirements. In this paper we argue the case for *pragmatic requirements* and extend the CGM with additional constructs to capture them and allow their analysis. We also develop an automated analysis which aids the planning and scheduling of tasks execution to meet pragmatic goals. Moreover, we evaluate our modelling and analysis regarding correctness and performance. Such an evaluation showed the applicability of the approach and its usefulness in aiding sensible decisions. It has also shown its capability to do so in a time short enough to suit run-time adaptation decision making.

1. Introduction

Adaptive systems are designed to enjoy a degree of flexibility in meeting their requirements and maintaining them. Adaptivity requires variability so that the system can choose amongst alternatives and optimize certain performance indicators. In requirements engineering literature, a main-stream model to capture and analyse such variability is Goal-Model (GM) [1]. It provides the requirements (goals) for which the system is designed and the various possible ways to reach those goals (alternatives) and their quality (soft-goals). It also allows breaking down the system to a set of autonomous entities (actors) who can also decide adaptively how to depend on each others to reach requirements.

Adaptivity is triggered by contextual factors which could be internal, e.g. errors and availability of computational resources, or external, e.g., newly available services and packages, physical and social environment of the user, their skills and computing device. In Ali el al. [2], traditional goal models [3–5] were extended to capture the notion of context and its relation with requirements. The proposed Contextual Goal Model (CGM) treats context as an adaptation driver which can help filtering the space of applicable alternatives to reach goals and dependencies between actors and deciding the quality of such alternatives and dependencies with

* Corresponding author.
E-mail addresses: felipepg@ime.usp.br (F.P. Guimarães), genaina@cic.unb.br (G.N. Rodrigues), rali@bournemouth.ac.uk (R. Ali), batista@ime.usp.br (D.M. Batista).

http://dx.doi.org/10.1016/j.datak.2017.03.003
Received 2 March 2017; Received in revised form 2 March 2017; Accepted 2 March 2017
0169-023X/

regards to certain quality measures (softgoals). The Runtime Goal Models proposed in Dalpiaz et al. [6] elaborate on specifying the possible valid alternatives to reach goals and the possible sequences for that achievement.

However, we advocate that requirements satisfaction is itself *pragmatic*. Pragmatism is the capability of solving problems in a sensible way that suits the current context rather than obeying fixed rules. This implies that goals refinements are not causal relations and the mere achievement of the sub-goals or execution of tasks does not necessarily imply that the parent goal has been achieved. There may be quality requirements to be achieved in order to consider a goal satisfied. Even more, such requirement can be lightened and strengthened dynamically depending on the context.

As an example, let's consider a scientifical paper's submission process. To have a paper accepted, one must have an idea, write about it, experiment on it and submit a paper. These steps could be seen as the decompositions of “Having a paper accepted” goal. However, performing these tasks would mean little if the submission's deadline is not met. This is a hard and clearcut quality criteria for achieving the root goal. Failing to meet this time constraint means failing the goal altogether, thus rendering it a pragmatic goal.

In Guimarães et al. [7], we have introduced the Pragmatic Goal Models (Pragmatic GM) embracing the concept of pragmatic requirements and pragmatic goals to grasp and model the idea that a goal's interpretation varies according to context. We have also developed an algorithm to compute if a goal is achievable under context-dependent quality of service (QoS) constraints and, if so, it returns a set of tasks that abide by such constraints.

In this paper we extend our model with goal annotations proposed by Dalpiaz et al. [6], which allows the Pragmatic GM to specify the runtime system behaviour. Now, instead of providing a simple *yes/no* goal achievability answer and applicable alternatives sets, through our new analysis and planning algorithm, the Pragmatic GM provides, through the new Pragmatic Planning Algorithm (PPA), a comprehensive goal fulfilment plan which achieves the pragmatic requirements and respects the allowable system behaviour at runtime. On top of that, we have conceived the new PPA algorithm certifying that it is still within a complexity suitable for runtime decision making. By these means, runtime adaptation of a software system benefits from our planning algorithm so as to schedule tasks execution and, therefore, meet the intended pragmatic goal. We evaluate our approach using a Mobile Personal Emergency Response System (MPERS) case study [8] and a scalability analysis. Results show that a plan for the MPERS case study was generated in less than a second. While scalability results point out that even for models with up to 10000 goals and 20 contexts sets the analysis and corresponding planning outcome can be reached within a minute.

The paper is organized as follows. Section 2 presents the CGM and RGM models on which the Pragmatic GM builds and extends. Section 3 presents the pragmatic goal and achievability concepts. Section 4 presents the proposed model, its component parts and the automated reasoning and plan engineering algorithm to find a suitable execution plan for the pragmatic model. Section 5 evaluates the applicability of our modelling and analysis approach. Section 6 presents related work and Section 7 concludes the paper and outlines our future work.

2. Goal models

Goal-Models (GM) are well established requirements engineering tools to depict and break-down systems using socio-technical concepts [1]. In other words, they provides the goals for which the system should be designed and the various possible ways to reach those goals. However, as pointed out in Ali et al. [2] and Dalpiaz et al. [6] three aspects of real-life cannot be captured in the traditional goal model [3–5]: the notion of contexts [2], the determination whether a task sequencing is valid within the model and the exploration of alternative system configurations to restore the system to a valid state [6].

2.1. Contextual goal-model

Contextual Goal Model, proposed in [2], are meant to capture the relation between requirements and their dynamic environment. Context can guide adaptation and support the decision in the goals to activate and filter the space of alternative strategies - subgoal, task or delegation - which can be applied to achieve activated goals. Context can also have an effect on the quality of those alternatives and this is captured through the notion of contextual contribution from goals and tasks to softgoals.

Context is description that concretize relevant factors in the system's environment, i.e., the surrounding in which it operates [9]. In goal modelling terminology, context is a specification of a partial state of the world relevant to an actor's goals [2]. Actors are social entities, or software representing them, in an organization. Actors exist to have and be responsible of achieving goals and they have degree of flexibility how to achieve them. A context that affects that choice could be the time of the day, a weather condition, patient's chronic cardiac problem, etc.

Fig. 1 present a CGM that depicts the requirements of a Mobile Personal Emergency Response System meant to respond to emergency situations for people in an assisted living environment. The root goal is “respond to emergency”, which is performed by the actor `Mobile Personal Emergency Response`. The root goal is divided into 4 subgoals: “emergency is detected”, “[p] is notified about emergency”, “central receives [p] info” and “medical care reaches [p]” ([p] stands for “patient”). Such goals are then further decomposed, within the boundary of an actor, to finally reach executable tasks or delegations to other actors. A task is a process performed by the actor and a delegation is the act of passing a goal on to another actor that can perform it.

It is important to highlight that not all the subgoals, delegations and/or tasks are always applicable. Some of them depend on certain contexts whether they hold.