2018 ACM/IEEE 40th International Conference on Software Engineering: Companion Proceedings

POSTER: How to Handle Environmental Uncertainty in Goal-based Requirements Engineering[1]

Manzoor Ahmad
LIUPPA
Pau, France
manzoor.ahmad@univ-pau.fr

Christophe Gnaho
LACL
Paris, France
christophe.gnaho@parisdescartes.fr

Jean-Michel Bruel
IRIT
Toulouse, France
bruel@irit.fr

Régine Laleau
LACL
Paris, France
laleau@lacl.fr

ABSTRACT

The objective of this poster paper is to investigate on how to deal with environmental uncertainty in goal based requirements engineering. To do so, we explore the introduction of RELAX concepts into SysMLKaos. RELAX is a Requirements Engineering language for Dynamically Adaptive Systems while SysMLKaos is a Goal based Requirements Engineering approach. We use an extract of a Landing Gear System case study to illustrate the proposed approach.

KEYWORDS

Dynamically adaptive systems, goal-oriented requirements modeling, environmental uncertainty

1 CONTEXT AND MOTIVATION

Complex and critical systems such as transportation systems can be considered as Dynamically Adaptive Systems (DASs) which are designed to continuously monitor their environment and then adapt their behavior in response to changing environmental conditions [1]. We believe that the success of the implementation of these systems depends, to a large extent, on Requirements Engineering (RE) approaches that explicitly capture the adaptability in requirements and takes into account the environmental uncertainty. However, few RE languages provide these constructs.

Goal Oriented Requirement Engineering (GORE) approaches are well suited to explore alternative requirements and can then be used to represent alternative behaviors that are possible when the system environment changes. Using GORE approach supposes that all possible alternative behaviors must explicitly be enumerated. However, changing environment factors makes it difficult to anticipate all the explicit states in which the system will be during its lifetime. GORE approaches are not thus sufficient to handle the uncertainty. On the other side, RELAX language [1] is a textual RE language that supports in contrast to GORE approaches, the explicit expression of environmental uncertainty in requirements. It provides a vocabulary to support the identification of the requirements that may be RELAX-ed, when the environment changes which includes a set of operators organized into modal, temporal, ordinal operators and uncertainty factors. However, unlike GORE, RELAX does not provide support for reasoning about alternative system configurations where different solutions can be explored and compared. In addition, it does not handle the way of operationalizing/realizing the RELAX-ed requirements. Consequently, we think that it would be interesting to take advantage of the contributions of RELAX by integrating it with a GORE approach.

We have proposed in our previous work SysMLKaos [2], a GORE approach that is based on two main ideas: to integrate Non Functional Requirements (NFRs) at the same level of abstraction as Functional Requirements (FRs) and emphasizing the impact of NFRs on FRs and to take advantage of the contribution of SysML[2]. In the meta-model of SysMLKaos, FRs and NFRs are represented as abstract goals, which are recursively refined into sub-goals, thanks to the AND/OR refinement mechanism. A Functional Goal (FG)/Non Functional Goal (NFG) that cannot be further refined into sub-goals is called elementary (FG or NFG). The objective of this poster paper is to investigate on how to incorporate RELAX concepts into SysMLKaos, and how to ensure that the RELAX-ed requirements will be realized.

2 THE PROPOSED APPROACH

The proposed approach includes three main steps that are respectively supported by three process models:

1. Identifying and expressing RELAX-ed requirements
2. Mapping RELAX-ed requirements to SysMLKaos
3. Applying SysMLKaos process to obtain an integrated goal model

2.1 Mapping between Relax and SysMLKaos

The mapping rules presented in this section is used in the second step to convert RELAX-ed requirements into SysMLKaos goal concepts. This activity is supported by a tool called RELAX2SysMLKaos editor [3].

ICSE '18 Companion, May 27-June 3, 2018, Gothenburg, Sweden

ACM ISBN 978-1-4503-5663-3/18/05.
https://doi.org/10.1145/3183440.3195013

[2] http://www.omgsysml.org/

In SysMLKaos, requirements are in the form of Abstract Goals while RELAX requirements are in the form of RELAX-ed (or invariant) requirements. In SysMLKaos, we express an elementary FG in the following format: Verb + Object where Object is related to properties that define the system's environment and Verb represents an action over this environment. For example: *Elementary FG1: Push (the verb) command (the object) to up* and *Elementary FG2: Close (the verb) the doors (the object).* In RELAX, the concept of ENV (environment) refers to properties that capture the operating context of the system. Therefore, this concept can be mapped to the SysMLKaos concept of Object. In SysMLKaos, an elementary FG is placed under the responsibility of an agent, a human being or automated component that are responsible for achieving the goal. In RELAX, the concept of MON (monitor) is used to identify properties that are directly observable and contribute information towards determining the state of the environment. For example: sensors, calculator, pilot etc. Therefore, the RELAX concept of MON can be mapped to the SysMLKaos concept of Agent. In RELAX, the concept of REL (relationship) is used to specify in what way the observables (given by MON) can be used to derive information about the environment (as given by ENV). Therefore, this concept can be mapped to the SysMLKaos concept Agent + Elementary FG (verb + object). For Dependency/Impact, SysMLKaos describes it as an Impact of an NFG on an FG. RELAX has positive and negative dependency which shows the dependency of a RELAX-ed requirement on another requirement.

2.2 Illustration

To illustrate the proposed approach, let us consider the following requirements extracted from the Landing Gear System (LGS)[3] (see the italic part of the sentence for the NFR part) : **FR "Retract Landing Gear" and NFR "Timed Response of the Landing Gear":** When the command line is working (normal mode), if the landing gear command handle has been pushed UP and stays UP then the gears shall be locked UP *before a maximum delay of 5 seconds after the handle position has been pushed up* and the doors shall be seen closed *before a maximum delay of 10 seconds after the gears locked up*. The potential identified environmental uncertainty for these requirements are shown in Table 1.

Table 1: Uncertainty factors for the retract landing gear

Uncertainty Factors	Description
ENV	Command, order, doors, gears
MON	Pilot, Calculator, Electro-valves, Cylinder
REL	Pilot pushes the command to up, Calculator treat the order, Electro-valves handles the command of the gears retraction, Cylinder do the gears retraction Cylinder do the doors closing/opening

The next step of the process consists in applying the RELAXToSysMLKaos mapping rules in order to convert the RELAX-ed requirements into SysMLKaos goal concepts. As a result, the above RELAX-ed requirement is first transformed to the following two abstract goals: ***Functional Goal*** [Retract Landing Gear] and ***Non-Functional Goal*** [TimedResponse (Landing Gear)]. Thanks to the SysMLKaos process, the two abstracts goals are refined in parallel, into two distinct goal models. This activity is partly guided by the environmental uncertainty information (especially for identifying the elementary FG and agents) and by the RELAX-ed expressions (useful to identify temporal constraints on some sub-goals). Finally, the impact of NFG on FG are identified and expressed. The result is a new goal model as shown in Fig. 1 that is called integrated goal model.

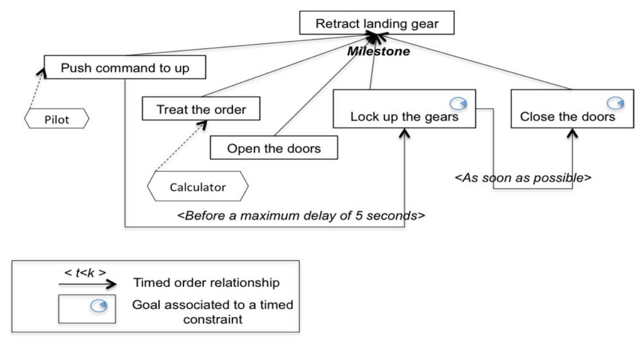

Figure 1: Integrated goal model

3 PERSPECTIVES

This is an ongoing work. We plan to verify some of these properties using formal methods like B. The overall objective is to bridge the gap between the requirements phase and the initial formal specification phase. We also plan to investigate the requirements dependencies in more detail i.e. once we decide to RELAX some requirements; what impact it induces on invariant requirements.

REFERENCES

[1] Jon Whittle, Pete Sawyer, Nelly Bencomo, Betty H.C. Cheng, and Jean-Michel Bruel. 2009. RELAX: Incorporating Uncertainty into the Specification of Self-Adaptive Systems," 17th IEEE International Requirements Engineering Conference, Atlanta, GA, 2009, pp. 79-88. doi: 10.1109/RE.2009.36.

[2] Christophe Gnaho, Farida Semmak, and Régine Laleau. 2014. Modeling the Impact of Non-Functional Requirements on Functional Requirements, In volume 8697 of the Lecture Notes in Computer Science series, 2014.

[3] Manzoor Ahmad, Jean-Michel Bruel, and Nicolas Belloir. 2015. Modeling and Verification of Functional and Non-Functional Requirements of Ambient Self-Adaptive Systems, The Journal of Systems and Software (2015), doi: 10.1016/j.jss.2015.05.028.

[3] https://www.irit.fr/ABZ2014/landing_system.pdf