

Using RELAX and KAOS for Adaptive Systems Requirements Modeling

Manzoor Ahmad and Jean-Michel Bruel

University of Toulouse

CNRS/IRIT

F-31062 Toulouse Université Cedex, France

Email: {manzoor.ahmad,bruel}@irit.fr

Abstract—Dynamic Adaptive Systems (DAS) modify their behavior at run-time in response to changing environmental conditions. For these systems, non-functional requirements play an important role, and one has to identify requirements that are concerned with the adaptive features. Goal based approaches can help in the development of requirements for DAS, keeping in view the inherent uncertainty in these systems. RELAX, which is a requirement engineering language, can introduce flexibility in non-functional requirements to adapt to any changing environmental conditions. This paper shows how to model adaptive systems requirements through an existing goal oriented approach that extends the SYSML meta-model and our proposed domain specific language for RELAX, that enables to derive requirements in graphical format from textual requirements in the form of SYSML requirements diagrams. Moreover, a merge of the two approaches will serve the development of DAS.

Keywords—Domain Specific Language; RELAX; Integrated Development Environment; Dynamic Adaptive Systems; Requirements Engineering;

I. INTRODUCTION

Most of the work in Requirements Engineering (RE) for Dynamic Adaptive Systems (DAS) assume that requirements already exists and their main focus is on requirements monitoring and reasoning about the correctness of adaptations [1]. The environmental conditions for these systems tend to change so they have to adapt to these changing conditions. In other words, we can say that DAS tend to be cyber-physical systems where the physical environment is tightly intertwined with the computing based system.

Goals can be used to systematically model the requirements of a DAS, they are treated as a collection of target systems with varying environmental conditions, so each target system's requirements are modeled and the adaptive logic that serves for the transition between the configurations are treated as separate concerns [2]. To develop goal models, a process can be used called LOREM (Levels Of Requirement Engineering for Modeling) [3], to represent the individual target system and the adaptive logic.

Our previous work [4] with requirements serve as a baseline for using RELAX [5] to model non-functional requirements. RELAX is a textual programming language which deals with uncertainty in DAS requirements which allows requirements to be temporarily relaxed to adapt to changing

environmental conditions. This relaxation is offered in case non-critical requirements be partially neglected in order to satisfy short term critical requirements. The distributed nature of DAS and changing environmental factors make it difficult to anticipate all the explicit states in which the system will be during its life time. As such a DAS needs to be able to tolerate a range of environmental conditions and contexts, but the exact nature of these contexts remains imperfectly understood. One overarching challenge in developing DAS, therefore, is how to handle the uncertainty posed by the respective application domains.

We can take benefits from goal oriented approaches in defining requirements for self adaptive systems. The worth of goal oriented techniques for requirements engineering in general can be found in literature [6] [7].

The context of our work is situated in Autonomic Computing where we are working on a case study of an AAL (Ambient Assisted Living) house [5]. The case study highlights the need to ensure patient's health in the AAL house. This house is equipped with an intelligent fridge that communicates with the AAL and is capable of reading, storing RFID information on food items. It also comes with 4 temperature and 2 humidity sensors. The fridge also detects the presence of spoiled food and discovers and receives a diet plan to be monitored based on what food items the patient is consuming.

The rest of the paper is organized as follows: section II shows the motivations of our work, background of the concepts, our previous experience, some challenges we are facing and the limitations of our work; section III shows why use goals for defining requirements, the integration of goals in defining requirements for DAS, their benefits and the relationship between different concepts and section IV concludes the paper with an insight on the future work.

II. OUR PREVIOUS WORK

A. Motivations

Our primary motivation is to enforce three aspects of the DAS requirements engineering:

- to ease the identification of those requirements on which the adaptation is going to apply (this is our primary work around RELAX [5])

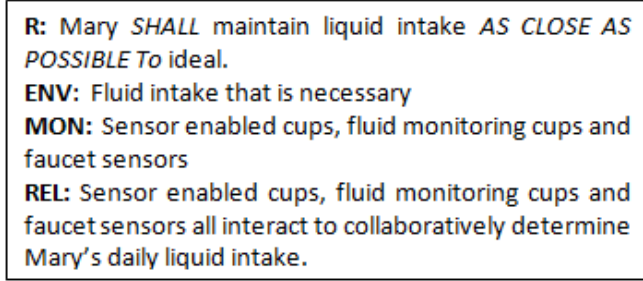


Figure 1. A requirement using the RELAX syntax [5]

- their traceability through the development life cycle (this is our primary work around SysML [8])
- to integrate goal oriented concepts in defining requirements for DAS

We have focussed so far on the requirements themselves (individually), the way we can write them in a more useful and precise way (see Figure 1), and the way we can automatically inject them in a system model (see Figure 2). The next step is now to work on the way we can identify the requirements that can be relaxed (called "relaxable") or not. This is why we are currently investigating the use of goal based approaches.

B. SysML

SysML¹ is a general purpose modeling language for systems engineering applications. It supports the specification, analysis, design, verification and validation of a broad range of systems and systems-of-systems. These systems may include hardware, software, information, processes, personnel, and facilities. It includes a graphical construct to represent text based requirements and relate them to other model elements. The requirements diagram captures requirements hierarchies and requirements derivation, and the satisfy and verify relationships allow a modeler to relate a requirement to a model element that satisfies or verifies the requirements. The requirement diagram provides a bridge between the typical requirements management tools and the system models.

C. RELAX

Typical textual requirements use modal verb SHALL that defines the functionality that a software system must always provide. RELAX takes the form of structured natural language, including operators designed specifically to capture uncertainty [9], their semantics is also defined. Uncertainty can be environmental and behavioral; environmental uncertainty is due to the changing environmental conditions such as sensor failure, noisy networks, malicious threats and unexpected human input, here uncertainty refers to maintaining the same requirements in unknown contexts

¹<http://www.sysml.org/specs/>

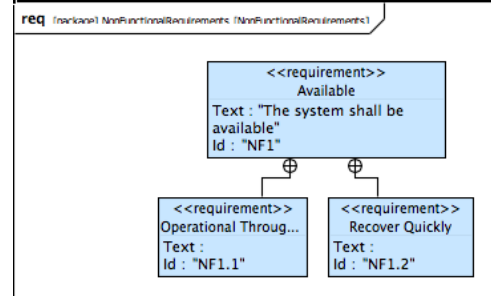


Figure 2. Generated Requirements Diagram

whereas behavioral uncertainty refers to situations where requirements themselves need to change.

The RELAX vocabulary helps in relaxing requirements when environment changes so it enables the analysts to identify the point of flexibility in their requirements. For this purpose RELAX process [5] is used which divides requirements into two types: variant or relaxed requirements that can be relaxed when the environment changes and invariant requirements that are fixed and cannot be changed keeping in view that it is the main functionality of the system. The relaxation also implies that a trade-off be made between critical and non-critical requirements when the environment changes, or resources are constrained, in such a case, we can compromise non-critical requirements for critical requirements.

In RELAX the conventional modal verb SHALL is retained and RELAX operators are introduced to provide more flexibility in how and when that functionality may be delivered. More specifically, for requirements that are left partially unsatisfied, the introduction of an alternative, temporal or ordinal RELAX-ation modifier will define the requirement as RELAX-able. These operators define constraints on how a requirement can be relaxed at run-time. In addition, it is important to indicate what uncertainty factors warrant a relaxation of these requirements, thereby requiring adaptive behavior. This information is specified using the MON (monitor), ENV (environment), REL (relationship) and DEP (dependency) keywords.

SysML incorporates requirements through requirements diagram so a link between SysML and RELAX would help in modeling the requirements efficiently. SysML provides a development environment and a graphical support for expressing all the variables of RELAX and also it helps in bridging the gap between the requirements and the overall system model as it is becoming an industry standard.

D. DSL for RELAX

Our previous work with RELAX is centered on a Domain Specific Language for self adaptive systems [4]. RELAX grammar is used as a meta-model for our DSL and based on this meta-model we would be able to bridge the gap between

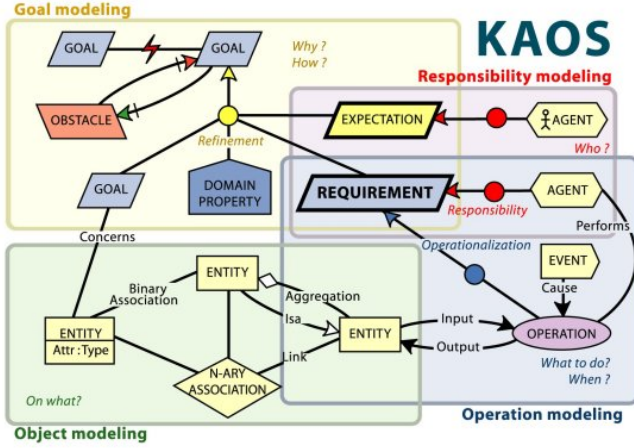


Figure 3. KAOS Models [12]

requirements and the overall system model.

Using our DSL [4], non-functional requirements in textual format are transformed into graphical format with the help of RELAX grammar in the form of requirements diagram. Figure 2 shows a snapshot of the generated requirements diagram.

For the generation of DSL, XText² is used. XText is a development framework for the development of DSL and other textual programming languages and helps in the development of an Integrated Development Environment (IDE) for the DSL. Some of the IDE features that are either derived from the grammar or easily implementable are: syntax coloring, outline view, model navigation, code completion and code templates. Among the benefits of XText, we have benefited from the code generation framework that is automatically generated from the grammar. A code generator has been written that is capable of processing models created with the DSL editor [8].

As a more perspective contribution a tool, COOL RELAX Editor [10] is developed using RELAX grammar as a meta-model. It is developed with all the RELAX variables embedded in it and with the help of this tool we can do M2T (Model 2 Text) and M2M (Model 2 Model) transformation.

E. SysML/KAOS

F. Current Limitations

III. OUR CONTRIBUTION IN REQUIREMENTS MODELING

A. The Overall Process

Figure 5 shows the integrated work plan.

An excerpt of the case study is given below: Mary is a widow. She is 65 years old, overweight and has high blood pressure and cholesterol levels. Following her doctor's instructions, she is considering to lose weight. The doctor

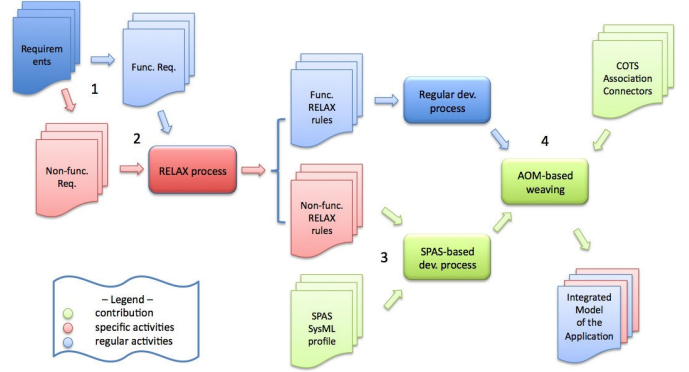


Figure 5. Integrated Work Plan

has recommended a hypo caloric diet with low levels of salt. She lives by herself in an AAL house.

For the correspondence between requirement and goal, we know that a `<<Block>>` satisfy `<<Requirement>>` and an `<<Agent>>` satisfy `<<Goal>>`, so we can deduce a hypothesis from this correlation that a `<<Requirement>>` corresponds to a `<<Goal>>` and a `<<Block>>` corresponds to an `<<Agent>>`. This hypothesis can be verified by the work of Chung [15] and Cysneiros [16] which shows that non functional requirements can be formulated in the form of goals.

Based on our experience with the AAL case study, we can deduce that requirement corresponds to goal. Figure 6 shows the SysML requirement diagram. Here the main requirement is to keep mary healthy; this requirement is composed of two requirements i.e. `<<Minimum liquid intake>>` which is satisfied by the block `<<Sensor enabled cups>>` and `<<Hypo caloric diet>>` which is satisfied by the block `<<Fridge display>>`. Figure 7 shows the KAOS responsibility model: here the main goal is to look after the mary diet plan which is decomposed into two sub goals. The first sub goal is the `<<Minimum liquid intake>>` which is achieved by the agent `<<Sensor enabled cups>>` and the other sub goal is `<<Hypo caloric diet>>` which is achieved by the agent `<<Fridge display>>`.

Uncertainty factors especially ENV and MON attributes are particularly important for documenting whether the system has the means for monitoring the important aspects of the environment. By collecting these ENV and MON attributes, we can build up a model of the environment in which the system will operate, as well as a model of how the system monitor its environment. Having said this, SysML/KAOS can complement RELAX by injecting more information in the form of positive/negative and direct/indirect impacts.

²<http://www.eclipse.org/Xtext/documentation>

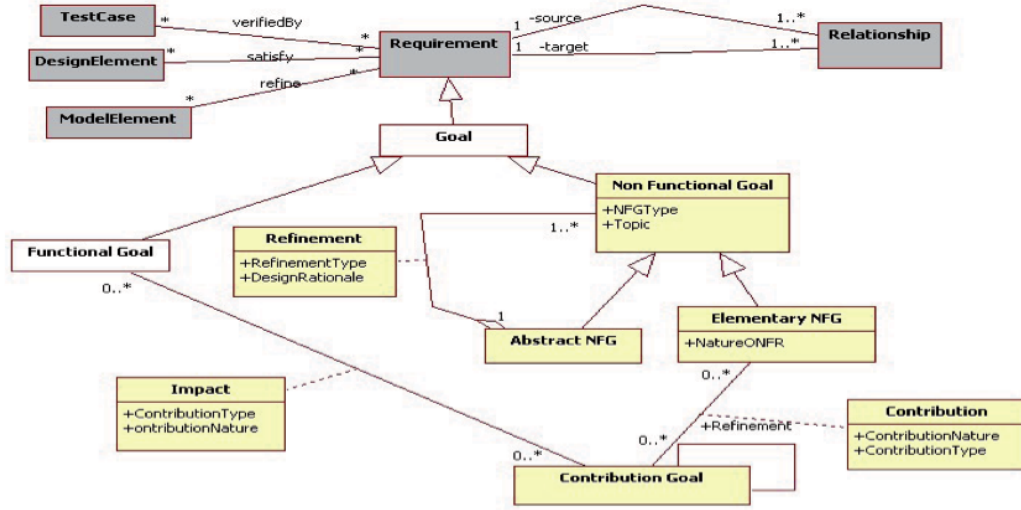


Figure 4. Extended SysML Meta-Model [13]

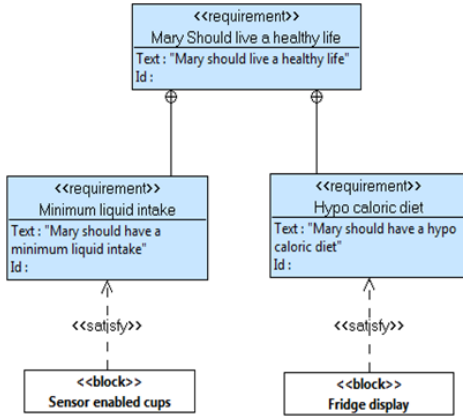


Figure 6. SysML Requirement Diagram

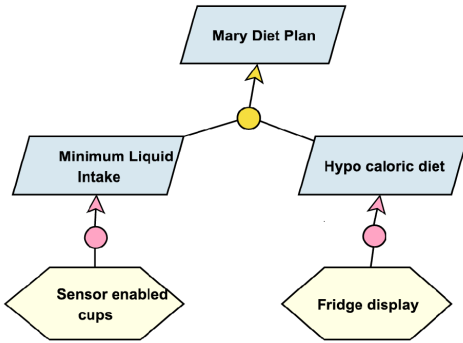


Figure 7. KAOS Responsibility Model

the concepts are not fully covered (e.g. `<<satisfy>>` for monitoring in SysML, this stereotype is used between a block and a requirement), but we have indicated in the table the closest mechanism that supports the concepts. In SysML/KAOS; requirements are described in the form of goals, SysML describes requirements in textual form while RELAX requirements are also in textual form with an enhanced version i.e. requirements divided into invariant and RELAX-ed requirements with uncertainty factors added to it. SysML/KAOS has no AND/OR refinement relationships, SysML has `<<verify>>` and `<<refine>>` relationships while for RELAX we have REL variable which identifies the relationship between ENV and MON. For Dependency/Impact SysML/KAOS deals with it as the impact of non-functional goal on functional goal; this impact can be positive or negative and direct or indirect while for SysML/ we have the concept of `<<derive>>` which shows the dependency between requirements, RELAX has positive and negative dependency. To deal with monitoring SysML/KAOS has the `<<contribution goal>>` concept which is used to satisfy a non-functional goal, SysML has `<<satisfy>>` which is used when a `<<block>>` satisfies a `<<requirement>>` while for RELAX we have the concept of MON which is used to measure the environment i.e. ENV. SysML/KAOS has a tool called SysML/KAOS editor, SysML has a number of tools e.g. eclipse³, papyrus⁴, topcased⁵ etc and for RELAX we have eclipse based COOL RELAX editor [10].

B. Why Use Goal Oriented Techniques for Requirements Engineering

C. Relationship b/w SysML/KAOS, SysML and RELAX

In Figure 8 we have studied how several key concepts are taken into account in the selected models. Most of the time,

³<http://www.eclipse.org/>

⁴<http://www.papyrusuml.org>

⁵<http://www.topcased.org/>

Concepts	SysML/KAOS	SysML	RELAX
Requirements Description	Goals	Textual Requirements	Enhanced Version of Textual Requirements
Relationship	AND, OR	<<verify>> <<refine>>	REL
Dependency/Impact	Contribution Nature: Positive Negative Contribution Type: Direct (Explicit) Indirect (Implicit) b/w NFG and FG	<<derive>> <<contain>>	DEP: Positive Negative
Monitoring	<<contribution goal>>	<<satisfy>>	MON
Tools	Eclipse based SysML/KAOS Editor	Eclipse/Papyrus/Topcased/	Eclipse based COOL RELAX editor

Figure 8. Relationship b/w Different Concepts

IV. CONCLUSION AND FUTURE WORK

Dynamic adaptive systems modify their behavior at run-time in response to changing environmental conditions. For these systems, non-functional requirements play an important role, and one has to identify requirements that are concerned with the adaptive features. For the development of requirements for DAS, goal based approaches play an important role keeping in view the inherent uncertainty in these systems. RELAX which is a RE language for self-adaptive systems can introduce flexibility in non-functional requirements; to adapt to any changing environmental conditions. This paper is based on requirements modeling using two different approaches: one through the use of existing goal oriented approach, for which SysML meta-model is extended with the goal concepts and the other proposed approach use RELAX which takes requirements in textual format and transform it into graphical format using RELAX grammar that acts as a meta-model.

We are of the view that SysML/KAOS can help RELAX inject more information in the form of positive/negative and direct/indirect impacts. Our work on RELAX is a part of an integrated work plan [17]. Our work and SysML/KAOS approach treat requirements from the very beginning and also the uncertainty that is surrounding self-adaptive systems. This work is a baseline for more concrete work as to explore fully the merging, the challenges related to this integration. In itself, integrating many modeling languages (KAOS, SysML and RELAX) is probably a good idea, as each of these languages brings its own analysis and has its own benefits. Another important aspect of RELAX is that the ENV, MON and REL attributes will be particularly interesting in building the SysML parametric diagrams so we can for example use mathematical equations to implement these attributes in the parametric diagram. Future work is centered around the development of the AAL case study. We are also interested in using formal methods to prove

that whether certain properties are complementary with each other or not even before development. As an AAL house contains different technologies ranging from medical services to surveillance cameras to intelligent devices e.g refrigerators etc., so using formal methods like B will greatly help in proving different properties.

ACKNOWLEDGMENT

The authors would like to thank their M.Sc. students for the COOL tool they have developed [10].

REFERENCES

- [1] B. H. C. Cheng, P. Sawyer, N. Bencomo, J. Whittle. A Goal-Based Modeling Approach to Develop Requirements of an Adaptive System with Environmental Uncertainty, MODELS '09 Proceedings of the 12th International Conference on Model Driven Engineering Languages and Systems Denver Colorado USA October 4-9, 2009, Springer-Verlag Berlin, Heidelberg 2009 ISBN: 978-3-642-04424-3.
- [2] J. Zhang, B. H. C. Cheng, Model-based development of dynamically adaptive software, ICSE 06: Proceedings of the 28th international conference on Software engineering, New York USA, ACM (2006) 371 - 380.
- [3] H. Goldsby, P. Sawyer, N. Bencomo, D. Hughes, B. H. C. Cheng, Goal-based modeling of dynamically adaptive system requirements, 15th Annual IEEE Int. Conf. on the Engineering of Computer Based Systems (ECBS) 2008.
- [4] M. Ahmad, First Step towards a Domain Specific Language for Self Adaptive Systems, 10th Annual International Conference on New Technologies of Distributed Systems (NOTERE), May 31- June 2 2010 Tozeur Tunisia, P. 285 - 290 ISBN: 978-1-4244-7067-9 DOI :10.1109/NOTERE.2010.5536629.
- [5] J. Whittle, P. Sawyer, N. Bencomo, B. H. C. Cheng, J. M. Bruel, RELAX: Incorporating Uncertainty into the Specification of Self-Adaptive systems, Proceedings of the 2009, 17th IEEE International Requirements Engineering Conference, Pages: 79-88 ISSN:1090-705X, 978-0-7695-3761-0.
- [6] Axel Van Lamsweerde, "Goal-Oriented Requirements Engineering: A Guided Tour", Fifth IEEE International Symposium on Requirements Engineering (RE'01) Toronto, 249-263.
- [7] Shahzad Anwer, Naveed Ikram, "Goal Oriented Requirement Engineering: A Critical Study of Techniques," 13th Asia Pacific Software Engineering Conference (APSEC'06), pp.121-130.
- [8] M. Ahmad, J. M. Bruel, Requirement Language Tooling with Xtext, UBIMOB 2011 7es Journées francophones Mobilité et Ubiquité Toulouse, Museum d'Histoire Naturelle 6 juin 2011 – 8 juin 2011.
- [9] J. Whittle, P. Sawyer, N. Bencomo, and B. H. C. Cheng. Reassessing languages for requirements engineering of self-adaptive systems, In RE Workshop for SOCCER08, Technical Report, 2008.
- [10] S. Gatti, J. Geisel, S. Labondance, J. Pages, COOL RELAX Editor, Internal Report M2ICE, Université de Toulouse II Le Mirail 2011.

- [11] A. V. Lamsweerde, Requirements Engineering: From System Goals to UML Models to Software Specifications. Wiley, 2009.
- [12] A KAOS Tutorial, V1.0 Respect-IT, 2007
- [13] Y. Belkaid, C. Gnaho, F. Semmak, Création de l'éditeur KAOS-SysML sous Topcased, Rapport Tacos, Juillet 2010.
- [14] http://www.omg.org/technology/documents/profile_catalog.htm
- [15] Chung et al., Non-functional Requirements in Software Engineering. Kluwer, 1999.
- [16] Conceptual Modeling: Foundations and Applications - Essays in Honor of John Mylopoulos (festschrift), LNCS volume 5600. Springer, 2009. 530 pp. ISBN 978-3-642-02462-7.
- [17] J. M. Bruel, N. Belloir, M. Ahmad, SPAS: un profil SysML pour les systèmes auto-adaptatifs, 15ème Colloque National de la Recherche en IUT (CNRIUT), Lille, June 8th to June 9th, 2009.