Towards validation of requirements models

Atif Mashkoor, Abderrahman Matoussi

Towards validation of requirements models

Atif Mashkoor[1] and Abderrahman Matoussi[2]

[1] LORIA – Nancy Université
Vandœuvre lès Nancy, France
{firstname.lastname}@loria.fr
[2] LACL – Université Paris-Est
Créteil Cedex, France
{firstname.lastname}@univ-paris12.fr

Abstract. The aim of this paper is to gradually introduce formalism in the requirement engineering phase in order to facilitate its validation. We analyze and elicit our requirements with KAOS, specify them into Event-B language, and then use the animation technique to rigourously validate the derived formal specification and consequently its semi-formal counterpart goal model against original customers' requirements.

1 Introduction

The use of formal methods for software development is escalating over the period of time. Most of the formal methods refine the initial mathematical model up to an extent where final refinement contains enough details for an implementation. The input to this formal specification phase is often the documents obtained during the requirements analysis activity which are either textual or semi-formal. Now there is a traceability gap between analysis and specification phases as verification of the semi-formal analysis model is difficult because of poor understandability of lower level of formalism of verification tools and validation of the formal specification is difficult for customers due to their inability to understand formal models.

Our objective is to bridge this gap by a gradual introduction of formalism into the requirement model in order to facilitate its validation. We analyse our requirements with KAOS (Knowledge Acquisition in autOmated Specification) [1] which is a goal-oriented methodology for requirements modeling, then we translate the KAOS goal model, following our derived precise semantics, into an Event-B [2] formal specification with the help of the platform RODIN[3], and finally we rigourously animate our specification with the help of the animator Brama [3], incorporated into platform RODIN, in order to validate the conformance of the specification to original requirements. With this approach we aim to reap benefits at two levels: customers can be involved into the development right form the start and consequently the requirement errors can be detected right on the spot.

[3] http://rodin-b-sharp.sourceforge.net

2 KAOS and Event-B

To analyze our requirements, we use KAOS which builds a data model in UML-like notation. The main KAOS goal defines an objective the system should meet, usually through the cooperation of multiple agents such as devices or humans, followed by several sub goals. Contrary to other requirements methods, such as i* [4], KAOS is well suited for our purpose because it can be extended with an extra step of formality which can fill in the gap between requirements and the later phases of development. The choice of Event-B as a formal specification language is due to its similarity and complementarity with KAOS. Firstly, Event-B is based on set theory with the ability to use standard first-order predicate logic facilitating the integration with the KAOS requirements model that is based on first-order temporal logic. Secondly, both Event-B and KAOS have the notion of refinement (constructive approach). Finally, KAOS and Event-B have the ability to model both the system and its environment.

3 Discussion

There are two main steps of our approach. First we translate our goal model into an Event-B specification with the help of our Event-B semantics, and later we animate the specification in order to validate that captured requirements are in accordance with original customer requirements. The whole process of validation is summed up by figure 1. Following is the brief elaboration of our approach:

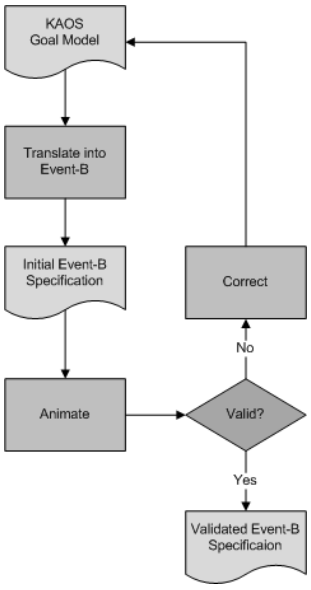

Fig. 1. The rigorous requirements validation process

3.1 The semantics step

The first step of our approach aims to express the KAOS goal model with Event-B by staying at the same level of abstraction which allows us to give this expression precise semantics. To achieve this objective, we use Event-B to formalize the KAOS refinement patterns that analysts use to generate a KAOS goal hierarchy. We primarily focus on most frequently used "Goal Patterns": the *Achieve goals*. The assertions in *Achieve goals* are expressed as following: $G\text{-}Guard \Rightarrow \diamond G\text{-}PostCond$, where *G-Guard* and *G-PostCond* are predicates. Symbol $\Rightarrow$ denotes the classical logical implication. Symbol $\diamond$ (the open diamond) represents the temporal operator "eventually" which ensures that a predicate must occur "at some time in future". Hence, such assertions demonstrate that from a state in which *G-Guard* holds, we can reach sooner or later to another state in which *G-PostCond* holds.

If we refer to the concepts of guard and postcondition that exist in Event-B, a KAOS goal can be considered as a postcondition of the system, since it means that a property must be established. The crux of our formalization is to express each KAOS goal as a B event, where the action represents the achievement of the goal. Then, we will use the Event-B refinement relation and additional custom-built proof obligations to derive all the subgoals of the system by means of B events. One may wonder whether the formalization of KAOS target predicates (i.e. the predicate after the diamond symbol) as B post conditions is adequate, since the execution of B events is not mandatory. At this very high level of abstraction, there is only one event for representing the parent goal. In accordance with the Event-B semantics, if the guard of the event is true, then the event necessarily occurs. For the new events built by refinement and associated to the subgoals, we guarantee by construction that no event prevent the post conditions to be established. For that, we have proposed Event-B semantics for each KAOS refinement pattern by constructing set-theoretic mathematical models. This process continues until the complete specification of KAOS goal model into Event-B. A detailed discussion on this step can be found in [5]. Formalization of the goal patterns other than *Achieve goals* is a work in progress.

3.2 The animation step

Following the precise semantics discussed in previous section, we derive an initial Event-B specification of the KAOS goal model. The aim of this animation step is to validate this derived specification. Our approach to address this issue is based on following hypothesis: we presume if the animation of the specification reveals the same behavior that we intended while writing our goal model, then the Event-B specification would be considered as a valid formal representation of the customers' requirements. In order to achieve this, we execute our specification to check its behavior with the approach defined in [6]. We rigourously animate the specification at each refinement step. It not only indicates any deviation from original requirements right on the spot but also helps fixing the specification errors. If any deviation from the intended behavior is discovered, we go back to the source and rectify the error. The process continues until the specification fully adheres to the requirements.

4 Conclusion and future work

We present an approach to validate a semi-formal requirement model by the animation of its formal counterpart. We express a KAOS goal model capturing users' requirements into an Event-B specification language for a stepwise requirement validation process.

At theoretical level our approach seems promising as we have obtained some initial results at its both steps independently. However, we hope that our proposed combined approach of analysis, specification and validation is also feasible collectively. We aim to target at transportation domain [7] to test our hypothesis.

5 Acknowledgements

This work has been partially supported by the ANR project TACOS (Ref: ANR-06-SETI-017). We would also like to thank Jean-Pierre Jacquot and Régine Laleau for their valuable comments.

References

1. Van Lamsweerde, A.: Requirements Engineering: From System Goals to UML Models to Software Specifications, Wiley (2009)
2. Abrial, J.R.: Modeling in Event-B: System and Software Engineering, Cambridge University Press (2009)
3. Servat, T.: BRAMA: A New Graphic Animation Tool for B Models, In: 7th International Conference of B Users (B'07), Springer-Verlag, Besançon, France (2007)
4. Yu, E.: Towards Modeling and Reasoning Support for Early-Phase Requirements Engineering, In: 3rd IEEE International Symposium on Requirements Engineering (RE'97), IEEE, Annapolis, USA (1997)
5. Matoussi, A.: Expressing KAOS Goal Models with Event-B, In Doctoral Symposium of 16th International Symposium on Formal Methods (FM'09-DS), Eindhoven, The Netherlands (2009)
6. Mashkoor, A., Jacquot, J.P., Souquières, J.: Transformation Heuristics for Formal Requirements Validation by Animation, In: 2nd International Workshop on the Certification of Safety-Critical Software Controlled Systems (SafeCert'09), York, UK (2009)
7. Mashkoor, A., Jacquot, J.P, Souquières, J.: B Événementiel pour la Modélisation du Domaine: Application au Transport, In: Approches Formelles dans l'Assistance au Développement de Logiciels (AFADL'09), Toulouse, France (2009)