Quantitative Assessment of Goal Models within and beyond the Requirements Engineering Tool: a Case Study in the Accessibility Domain

Christophe Ponsard[1] and Robert Darimont[2]

[1] CETIC Research Center, Charleroi (Belgium) christophe.ponsard@cetic.be
[2] Respect-IT SA, Louvain-la-Neuve (Belgium) robert.darimont@respect-it.be

Abstract. Goal-Orientation provides a rich framework to reason about systems at Requirements Engineering (RE) time, at least using some quantitative form of reasoning on a goal structure. This paper focuses on the assessment of the satisfaction level of requirements and related goals which has to be measured at run-time and possibly involves some multiple instantiation schemes. We propose a framework to derive such a run-time assessment from the design-time goal model and illustrate it on a real-world case study from the accessibility domain.

1 Introduction

Goal-Oriented Requirements Engineering (GORE) approaches such as KAOS [7], i* [17], URN/GRL [6] rely on a rich meta-model guiding the requirements engineer to systematically capture relevant aspects of the problem. They have also shown their value to reason on requirements and ensure key requirements qualities such as completeness, consistency and robustness.

Over the years GORE frameworks evolved towards more powerful forms of reasoning. While some formal approaches have been considered (like FAUST [10] or Formal-TROPOS [5]), the formalisation overhead is often too heavy for the majority of systems. Both as an alternative and a complement, more focused approaches supporting some kind of quantitative reasoning have been proposed by most of the GORE frameworks. For example, reasoning mechanisms on partial/probabilistic satisfaction on goal and obstacle models are described in [3, 8], and different propagation algorithms are detailed by [1]. More recently, the use of domains specific Key Performance Indicators (KPI) has also been introduced [2]. On the tooling side, all major GORE tools are also model-based and support quantitative reasoning, e.g. jUCMNav (GRL) [16], Objectiver (KAOS) [12], RE-tool (multi-notational extension to StarUML) [14].

Such tools are design time. However, goal assessment via tools is often required after design time, typically in auditing processes or even when considering the KPI dashboards used in business intelligence from a goal perspective. This paper aims at detailing a run-time approach to measure goal satisfaction based on a quantitative assessment model that has to deal with a concrete instantiation level. Our purpose is not to propose a generic framework but rather illustrate how we successfully transferred design-time reasoning at run-time on a convincing real-word case study in the assessment of the physical accessibility of public places.

The paper is structured as follows: Section 2 gives an overview of our approach. Then, Section 3 details our case study. Finally, Section 4 concludes with our current status and roadmap.

2 Proposed Approach and Reference Implementation

Fig. 1 shows specific activities occurring at the RE and assessment times:

- **At RE time.** The RE analyst builds a RE model decorated with quantitative assessment rules covering all strategic goals. It is also important to capture the domain structure as goals might be instantiated multiple times (e.g. for emergency evacuation plan, all rooms must be assessed).
- **At assessment time**. The domain expert typically uses a form-based assessment tool. A spreadsheet can be used if there is either a limited form or no form of instantiation. Otherwise, a domain specific tool must be considered and can actually be derived from the model.

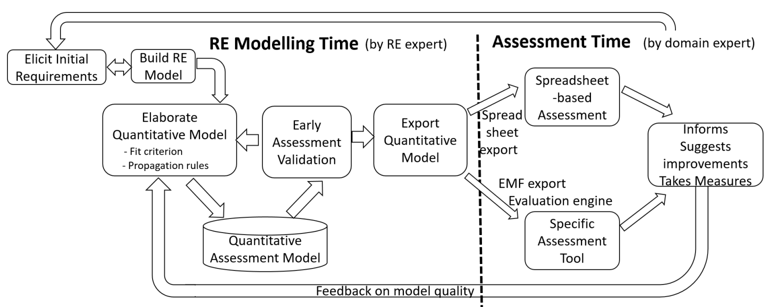

Fig. 1. Our Quantitative Framework

Our implementation relies on the Objectiver GORE tool [12] which is based on KAOS [7]. However, our approach can be transposed to all major GORE methods and tools mentioned earlier. Although, our approach can benefit from reasoning on obstacle which is more elaborated in KAOS. This tooling provides rich integration of models, diagrams and tables [9]. Run-time assessment requires the following key steps:

1. Know how to assess a goal in a specific context: this relies on propagation rules from requirements up to a goal that are typically specified in goal models. However, the model needs to be enriched with specific context to identify the involved instances. In our case, to assess an entrance, we need to identify one or more doors that can be used to enter.
2. Know the state of the instance: this is the data that are actually collected from the field. Note that if it is manually collected, it is worth revealing the underlying goal to the assessor to help him understand his work. In our case,

assessing the door means looking at how easy it is to open it, if there is a doorstep, in case of a transparent door check if it has marks, etc.
3. Know how to combine multiple instances of a goal. For this, domain specific rules are required. In our case: assessing alternative entrances will result in keeping the entrance with the maximal accessibility while assessing multiple physical barriers on an entrance will result in keeping the worst barrier. Such rules will be detailed later.

Achieving run-time model instantiation and goal assessment can be achieved through the following tooling:

- Directly use the provided instance level of the tool. However, this is very impractical to use and is only recommended for early model validation.
- Generate an assessment spreadsheet from the model in which the assessment logic from measurable requirements up to goal-level KPI can be transposed. This can however only be achieved for domains where the instantiation is statically known (for example: 3 offers will be ranked in a call for tender).
- Export the model together with the propagation rules as a run-time module that can be evaluated on multiple instances. This most general case is required for dynamic instantiation like in our case study where each building can have different number of rooms, services, equipments, etc. We rely on the widely used Eclipse Modelling Framework (EMF) for this purpose [13].

3 Case Study - Accessibility Assessment of Public Infrastructures

Assessing the accessibility of public places for mobility-impaired people requires to capture measurable accessibility requirements. These must actually address a number of (physical) obstacles to reach the objective of a person's visit. A goal/obstacle model enriched with quantitative evaluation criteria is very relevant and has been developed for Belgian public authorities [11]. The model is structured around 4 main accessibility goals, 50 physical elements, 6 different disability profiles. The analysis resulted in about 185 evaluation rules either with direct measurement points (requirements) or aggregation rules. In order to support the assessment, a form-based environment was developed and configured using the EMF export of the GORE model.

The target users are mobility impaired people with the key accessibility requirements related to their (dis)abilities. Fig. 2.a shows the top level which is structured using milestone patterns representing the physical progression towards the accomplishment of the purpose of a visit: (1) reach building/parking, (2) cross entrance and reach welcome desk, (3) circulate to reach target equipment/service, and (4) make use of target equipment/service. Those are refined until reaching accessibility requirements on elementary actions such as opening a door, going upstairs, etc.

In order to discover fine grained requirements, a full obstacle analysis is carried out on them. Fig. 2.b shows a limited excerpt of that analysis related to

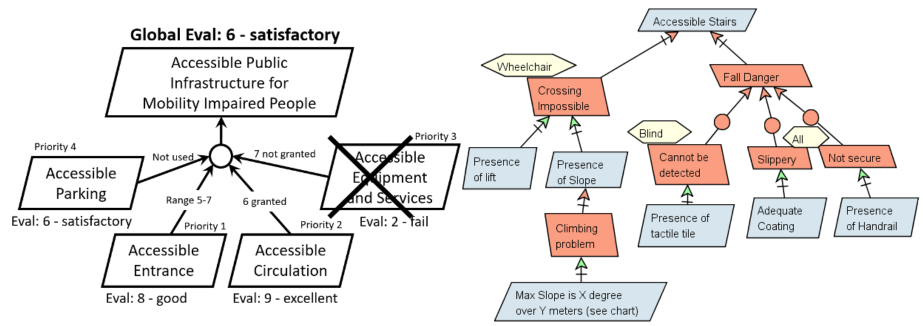

Fig. 2. (a) top level goals and (b) obstacle analysis

stairs. The satisfaction measure is related to the degree of absence of any obstacle and depends on the target user: e.g. for a wheelchair, the presence of stairs can be a problem while for blind people it is the absence of security signaling which is a problem. Alternatives are also taken into accounts, e.g. the presence of a lift or slope next to a stair will restore accessibility for a wheelchair. Quantification is achieved by measuring presence or specific characteristics of elements (e.g. maximal allowed slope as a function of the distance to cover).

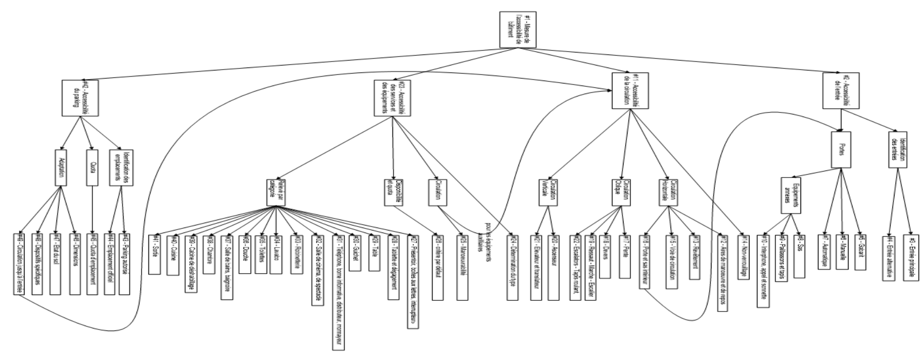

Fig. 3. Overview of the complexity of the accessibility goal-model

The resulting assessment model covers about 50 accessibility requirements for each category. Its global structure is depicted in Fig. 3, structured over 3 to 4 layers starting from the 4 main milestones stated earlier which are further decomposed into finer grained milestones, e.g. horizontal, vertical and oblique circulation are distinguished. Satisfaction of higher level accessibility goals must also be defined using specific rules for intermediary goals. Those relate to the way the elements instances are combined. Parallel (alternatives) will typically keep the maximally accessible element while milestone will keep the minimally accessible element. In reality, rules are more complex because different accessibility levels are distinguished. So, more complex arithmetic combining accessibility levels of child goals come into play to define the accessibility level of the parent goal. For example, the formalisation of the rule for entrance is:

$$Measure(GlobalAccessibilityOfEntrance) = max\{altEntrance : Entrance \mid min(\\ Measure(AccessibilityOfWayToEntrance(altEntrance)),\\ Measure(AccessibilityOfEntrance(altEntrance))\ )\ \}$$

As the goal model is generic and cannot cope with multiple instances of specific elements, the model needs to be exported in an external tool able to instantiate all relevant elements based on the domain structure, capture their organisation (e.g. how many alternative entrances), assess them according to leaf criteria (e.g. accessibility of door and of way to door) and propagate the accessibility score using the defined rules. The form interface is illustrated in Fig. 4 with the structure on the left part and the form on the right part.

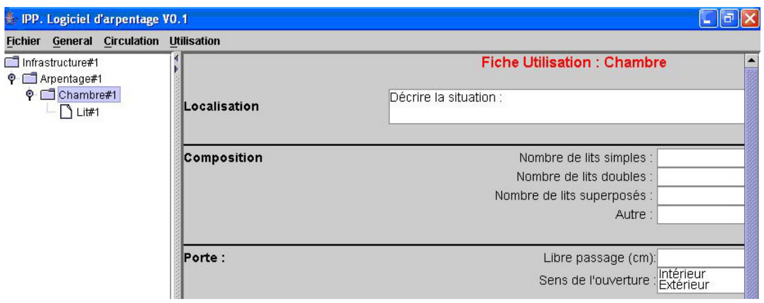

Fig. 4. Generated form using EMF model (in French)

Based on the collected information, a fully objectified assessment report can be produced. It provides an overall picture of the accessibility by user category. After validation, the assessment is published in an easy to interpret form on the access-i website as shown in Fig. 5 together with a summary of strongest and weakest points. Easy to interpret codes are used like grey, orange, green and symbols like --,-,0,+,++.

Fig. 5. Example of published accessibility information summary

4 Discussion, Conclusion and Future Work

We demonstrated how to use a mainstream RE tool to support a set of primitives for performing quantitative evaluation not only inside the tool itself but also using external and independent tools. A benefit of the approach is that domain experts can more easily perform high quality quantitative analysis relying on strong rationales. The process also provides a feedback loop from the domain expert back to the RE analyst, so the model can be evolved on the long run. This evolution has actually happened: the inital "Master Key Index" method helped in merging different methods used in Belgium under the "access-i" method which is now increasingly being recognised and used abroad [4].

At this point, the generation framework is still ad-hoc. Although propagation is supported, the main limitation is about the instantiation step over the domain structure which is not yet automated. For spatial structures, we have recently introduced relevant extensions borrowed from Geographic Information Systems [15]. Those provide the ability to characterise model elements with spatial nature and relationships among them (part-of, next-to, etc.), and express the required quantified properties on them.

References

1. Amyot, D., et al.: Evaluating goal models within the goal-oriented requirement language. Int. J. Intell. Syst. 25(8), 841–877 (Aug 2010)
2. Barone, D., Jiang, L., Amyot, D., Mylopoulos, J.: Reasoning with key performance indicators. In: The Practice of Enterprise Modeling - 4th IFIP WG 8.1 Working Conference. LNBIP, vol. 92, pp. 82–96. Springer (Nov 2011)
3. Cailliau, A., van Lamsweerde, A.: Assessing requirements-related risks through probabilistic goals and obstacles. Requir. Eng. 18(2), 129–146 (2013)
4. CAWAB: Access-i Physical Accessibility Portal. http://www.access-i.be
5. Fuxman, A., et al.: Specifying and analyzing early requirements in tropos. Requir. Eng. 9(2), 132–150 (May 2004)
6. International Telecommunication Union: Recommendation Z.151 (10/12), User Requirements Notation (URN) - Language Definition (2012)
7. van Lamsweerde, A.: Requirements Engineering: From System Goals to UML Models to Software Specifications. Wiley (March 2009)
8. Letier, E., van Lamsweerde, A.: Reasoning about partial goal satisfaction for requirements and design engineering. SIGSOFT Soft. Eng. Notes 29(6) (Oct 2004)
9. Ponsard, C., Darimont, R., Michot, A.: Combining Models, Diagrams and Tables for Efficient Requirements Engineering: Lessons Learned from the Industry. In: INFORSID, Biarritz (June 2015)
10. Ponsard, C., Massonet, P., Molderez, J., Rifaut, A., van Lamsweerde, A., Van, H.T.: Early verification and validation of mission critical systems. Formal Methods in System Design 30(3) (2007)
11. Ponsard, C., Snoeck, V.: Objective accessibility assessment of public infrastructures. In: 10th Int. Conf. ICCHP, Linz, Austria. LNCS, vol. 4061 (July 2006)
12. Respect-IT: The Objectiver Goal-Oriented Requirements Engineering Tool. http://www.objectiver.com (2005)
13. Steinberg, D., Budinsky, F., Paternostro, M.: EMF: Eclipse Modeling Framework (2nd Edition) (2008)
14. Supakkul, S., Chung, L.: The re-tools: A multi-notational requirements modeling toolkit. In: 20th IEEE Int. Req. Eng. Conf. (RE) (Sept 2012)
15. Touzani, M., Ponsard, C.: Towards modelling and analysis of spatial and temporal requirements. In: 24th IEEE Int. Req. Eng. Conf., Beijing, China, Sep. (2016)
16. University of Ottawa: jUCMNav: Juice up your modelling. http://jucmnav.softwareengineering.ca/twiki/bin/view/ProjetSEG (2001)
17. Yu, E.S.K., Mylopoulos, J.: Enterprise modelling for business redesign: The i* framework. SIGGROUP Bull. 18(1), 59–63 (Apr 1997)