Goal-oriented Decision Modeling: A Position Paper

Renata Guizzardi[0000-0002-5804-5741]1 and Anna Perini [0000-0001-8818-6476]2

[1]Ontology and Conceptual Modeling Research Group (NEMO),
Federal University of Espírito Santo, Brazil
[2]Fondazione Bruno Kessler, Italy
rguizzardi@inf.ufes.br, perini@fbk.eu

Abstract. The Decision Modeling Notation (DMN) has been proposed by OMG as a means to model decisions, such as those that can take place in business processes, thus, for instance, supporting business analysts to analyze them and possibly derive specification for their automation. We are interested to investigate if and how DMN could be complemented by goal oriented (GO) modelling with the purpose of making more explicit how decisions can affect intentional elements, e.g. their impact on satisfying or not goals, as well as allowing a more concrete documentation of decisions, together with their rationale. In this paper, we introduce a method that allows combining *i** and DMN models, discuss challenges and opportunities we identify while developing it.

Keywords: Decision-Making, Goal, Conceptual Modeling Method.

1 Introduction

Decision-making refers to agents' choices taken on the basis of their knowledge, including beliefs and preferences [1]. It is an essential part of the everyday life of any individual and organization. On the one hand, decision-making serves very simple purposes, such as how one likes to take one's coffee, but within an organizational environment, even very simple decisions, for which no one has given too much thought, can have a profound impact on the way things turn out.

Decision-making has been the focus of many research fields within Computer Science, such as Artificial Intelligence, Software and Requirements Engineering, and Business Process Analysis, with different purposes, ranging from automating it, developing decision support systems at use of a human decision-maker, integrating decision-making in a flexible way into business processes. Despite all the interest in decision-making, both in academic and enterprise settings, no definitive method is today accepted for modelling and documenting decisions. Focusing on that, OMG has recently proposed a language named Decision Modeling Notation (DMN) [2], which provides business analysts with the constructs for modelling decisions, depicting them in simple diagrams, and possibly deriving specification for their automation. DMN has been integrated with the Business Process Modeling Notation (BPMN) [3], to make explicit when decisions are associated to tasks.

We are interested to investigate if and how DMN could be complemented by goal oriented (GO) modelling, by making more explicit how decisions can affect intentional elements, e.g. their impact on satisfying or not goals, as well as allowing a more concrete documentation of decisions and tracing decisions to intentional elements. As a first step in our research [4], we analyzed the semantic correspondence of notation elements in a GO modelling notation, namely *i** [5], and in DMN. Building on this semantic mapping, in this paper, we propose a method that combines *i** and DMN with the purpose of supporting a smooth transition from *i** to DMN models, which can be possibly automated.

In the rest of this short paper, we mention relevant related work on decision modeling (Section 2), introduce the proposed goal-oriented decision modelling method (Section 3), and conclude discussing challenges and opportunities we identified, as well as next steps in our research.

2 Decision Modeling

Among research works supporting decisions' representation and reasoning, we mention here those which are motivating and influencing our research on decision modelling. For example, the works about decisions on goals in KAOS [6] and in the Business Intelligence framework [7]. Lamsweerde [6] proposes the use of decision tables to represent the options related to goal achievements in an OR goal decomposition. The tables allow to perform both qualitative and quantitative reasoning about the best choice of goals achievement, given some quality criteria to be ensured. The Business Intelligence framework [7] proposes the use of GRL modelling language (based on *i**) to perform a quantitative reasoning on goal achievements, thus supporting the decisions related to the best strategies to pursue a given business objective. Both these proposals add elements (such as tables, or business-related parameters) either as tools not integrated in the modelling language or as extension of the modelling language (in the case of the BI proposal). On the contrary, our proposal intends to bridge and align two complementary languages using ontologies.

Focusing on decisions in the software engineering domain, such as selecting architectural assets for software-intensive systems, Papatheocharous et al. [8] propose a taxonomy called GRADE, which has been defined involving ten experts on decision-making in such domain. The taxonomy has been validated on real case studies. Among its main purposes are on one hand supporting practitioners in structuring their decisions, and on the other hand allowing to document decision-making processes and to classify and compare them along their outcomes. Key elements proposed in GRADE for characterizing decisions are: i) Goals, ii) Roles, iii) Assets, iv) Decision methods and criteria, and v) Environment. We believe that the GRADE's elements can be used to model decisions not only in the software engineering domain, but also to provide support towards a goal-oriented approach for decision model.

3 Proposed Method

The proposed method can be described in terms of an iterative lifecycle composed of the following phases (in execution order): *Goal analysis* (composed of the *Goal elicitation* and *Goal modeling* sub-phases), *Identifying decision elements*, and *Decision*

Analysis (composed of these four sub-phases: *Automatically Generating DRD*, *Refining DRD*, *Automatically Generating Decision Tables* and *Refining Decision Tables*).

Goal analysis proceeds as usual, with goal elicitation and modeling, and eventually, other steps necessary to model a stable set of goals, such as goal negotiation and conflict management. Consider as a goal analysis' result, the strategic rationale *i** model of Fig. 1, which has been created synthetizing real cases explored in an existing European project[1] and is used to illustrate the proposed method in the remainder of this paper.

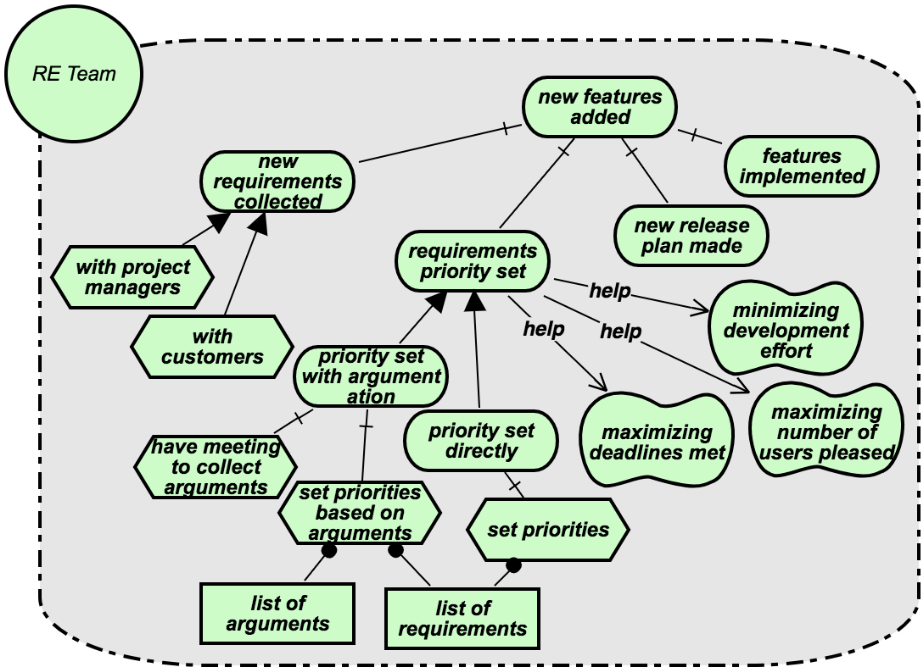

Fig. 1. *i** Strategic Rationale Model of the Requirements Prioritization Example

The model describes the goals of a Requirements Engineering Team (see RE Team[2] actor in Fig. 1) w.r.t a main goal of having new features added to an existing software product. To accomplish this main goal, the RE Team must accomplish four subgoals, namely: new requirements collected, requirements priority set, new release plan made ***and*** features implemented. For reasons of space, the model is not described any further in this section, but we hope the meaning of its remaining elements becomes clear as long as we use them as examples throughout the paper.

This section proceeds as follows: subsection 3.1 describes the *Identifying decision elements* phase, while subsections 3.2 and 3.3 focus on the sub-phases composing *Decision Modeling*.

---

[1] SUPERSEDE (www.supersede.eu) is an H2020 project aiming at providing methods and tools to enable data-driven software evolution and dynamic adaptation.

[2] Throughout the paper, we use a different font to support the identification of the *i** model elements (i.e. actor, goals, quality goals, tasks, and resources).

3.1 Identifying decision elements

After goal analysis comes the step aimed at identifying decision elements in the goal model. In our method, we consider four elements which can be understood according to the GRADE taxonomy [8] definitions of *role*, *goal*, *criteria* and *asset*, namely, the *i** concepts of *actor*, *goal*, *quality goal* and *resource*, respectively.

The **actor(s)** defines the perspective along which a decision needs to be taken, that is an individual or an organization, a product owner or a political actor, and the actors who may contribute to reach a decision. We believe this element corresponds to the GRADE decision element called *Role,* which is defined as the way to characterize decision-makers along different attributes, and also to clarify their contribution to the decision made [8]. In our example, the RE Team actor (see Fig. 1) is the focused decision-maker.

An important part of this phase is identifying the decision-maker **goal(s)**, i.e. the target or desired state of the world that requires a decision (called from now on, *decision goal*). In *i** models, we use OR-refinement to model alternative goals to be pursued or tasks to be executed. In other words, a decision must be made whether to take the left or the right branch of the OR-refinement tree. Thus, this kind of link can be seen as an indicator of a decision goal, which may be straightforwardly identified (i.e. the parents in an OR-refinement goal tree are considered decision goals). For instance, in Fig. 1, the requirements priority set goal may be accomplished by the achievement of either priority set with argumentation or priority set directly. According to our method, the requirement priority set is hence a decision goal.

The **criteria** that a decision aims at optimizing are usually modelled in *i** as quality goals related to an OR-refinement tree (i.e. directly or indirectly linked to a decision goal). In our example, the RE Team prioritizes requirements aiming at minimizing development effort, maximizing number of pleased users and maximizing deadlines met (modeled as quality goals in Fig. 1). These quality goals are, thus, the criteria identified in our example.

The assets that are necessary to perform a decision are usually modelled as **resources** or other intentional elements in means-end relationships with elements in a OR-refinement tree. As an illustration, take the list of requirements resource of Fig. 1.

We highlight that goals that are not parents in an OR-refinement tree may also be identified by the analyst as goals that motivate decision-making in order to be operationalized. As an example, take the new release plan made goal of the model of Fig. 1. Intuitively, we understand that making a plan certainly requires a lot of decision-making, even if in our model, the decision alternatives are not explicit. In fact, identifying such decision goals may indicate points which require the goal analysis to be deeper and the goal model to be refined.

3.2 Generating and Refining Decision Requirement Diagrams

In DMN, decision analysis starts by creating Decision Requirement Diagrams (DRD), depicting the relation between distinct decisions and what kind of information, document and/or actor is necessary for that decision to be made. In our proposed method, draft diagrams may be even automatically generated or suggested to the analysts, who can then refine them (see next section for DRD refinement).

For the automatic generation, this work takes a model-driven approach, in which *i** modeling elements are mapped into DMN modeling elements. By taking this approach, we aim at facilitating automatic mapping from one language to the other, whenever possible. Besides that, we take into account the semantics of the modeling elements of each notation, mapping them to a decision-making ontology. An *i** modeling element corresponding to the same ontological concept of a DMN element will consequently be mapped to each other. By taking this semantic strategy, we aim at avoiding the pitfall of mapping modeling elements only based on their label, which may be misleading.

Table 1 shows the mapping rules used to map an *i** model into a DMN model. From this table, all lines but the last refer to this particular phase. The decision ontology and the rationale for each mapping rule are describe in [4].

Table 1. *i** to DMN Mapping Rules

i* Element	**DMN Element**	**Ontological Concepts**
OR-refinement parent goal	decision	goal
goals pertaining to the same OR-refinement tree	decision dependency	goal, deliberative act, decision, criterion
i* resource connected to goal pertaining to an OR-refinement tree	knowledge source (document)	resource
	input data	belief's propositional content
actor A whose rationale is being analyzed or in a dependency relation with A	knowledge source (person)	agent
quality goal (sometimes hardgoal) related to a goal pertaining to an OR-refinement tree	criteria to be refined in decision tables	goal

According to our method, a draft DRD is automatically generated and then, it must be refined by the analyst to adjust the elements' names and to include further elements and relations if needed. Perhaps in decision modeling more than in any other kind, human intervention is highly recommended, as it is during such opportunities that decisions are actually taken.

As already mentioned in section 3.1, OR-refinement goals are special candidates to become decision goals and as such, automatically generate decisions in a DRD. For instance, in our example, the DRD would be composed of three decisions (first automatically identified and then, having their names adjusted by the analyst): requirements prioritization approach definition to model the decision of whether the requirements prioritization would require argumentation or would be accomplished directly, priority set with argumentation and priority set directly, both to model the requirements prioritization decision itself, following each approach. Moreover, there would be a dependency link between each of these two later decisions and the former, to model that deciding on the requirements' priorities depend on deciding the requirements prioritization approach in the first place.

The input data used for decision-making is also depicted in the DRD. For instance, the resource identified in the previous phase (e.g. list of requirements) would be automatically depicted as an input data in our DRD. Moreover, in case there were other resources containing prioritization criteria or actors on which the RE Team depended to make a decision, these resources and actors would be depicted as knowledge sources in the DRD.

By showing dependencies between decisions, this diagram helps one to analyze in which order decisions should be taken, and to estimate the complexity of a particular decision. Moreover, by depicting the knowledge source, i.e. particular documents or actors that should be consulted to make a decision, a DRD assists vulnerability analysis, in the sense that if many decisions are dependent on a particular actor of an organization, this shows a high dependency of the organization on that person, who will be terribly missed in case he/she leaves. These are only a few examples of the kinds of analysis supported by DRDs.

3.3 Generating and Refining Decision Tables

Maybe the most crucial model proposed by DMN is the decision table, in which the actual rationale behind decisions may be documented. In other words, each line of a decision table shows what specific decision is taken, given a number of criteria. This allows people reading the table to replicate such decisions given similar contexts. Suppose, for instance, an organization's newcomer, who may learn how to proceed by looking at the documented decisions.

In $i*$ models, contribution analysis using quality goals is often used to support decision-making. Our method proposes that the criteria in the decision table come from such quality goals (see Table 1, last line). Thus, a decision table automatically generated from our example would present three criteria, namely: minimizing development effort, maximizing number of users pleased, maximizing deadlines met. Then, the analyst would be responsible for refining the automatically generated decision tables. In particular, we must say that although quality goals may lead to a decision criterion, a lot of work remains to choose a proper criterion name, the criterion's possible values, and at last, to define the outcome of the decision, given the set of criteria. Take for instance Table 2, which presents a possible decision table for our example, after such refinement.

Table 2. Decision Table of the Requirements Prioritization Example

U	Number of users negatively affected $< 50, \geq 50$	Number of users positively affected $< 500, \geq 500$	Feature development time $\leq 20, > 20$	Requirement priority set $high, medium, low$
1	< 50	≥ 500	≤ 20 $man\ hours$	$high$
3			> 20 $man\ hours$	$medium$
4		< 500	-	low
5	≥ 50	-	-	low

The first and second columns represent two criteria based on maximizing number of users pleased; and the third column, a criterion based on minimizing development effort. For reasons of space, we did not consider the maximizing deadlines met quality goal when creating this decision table. By examining each table line, one can understand what decision is taken w.r.t requirements prioritization (see last column), based on the modeled criteria.

4 Discussion and concluding remarks

This short paper proposes a novel method to document decisions by integrating GO (e.g. *i**) and DMN models. This method is based on a semantic mapping between some of the entities offered by the two languages, defined in an earlier work [4].

Our research builds on the idea of trying to exploit the different characteristics offered by the two modelling languages. While GO models, as *i** models, allows specifying and analyzing decisions by explicitly modelling the decision-maker's perspective, goals, criteria and available assets, DMN offers modelling entities at a lower level of abstraction which can be useful to specify how decisions can be operationalized. Using a GO approach in early phases of decision modelling can also provide the opportunity to reason on alternative decision-making methods, e.g. based on expert knowledge or on data analytics.

Several aspects need to be further investigated towards consolidating the proposed goal-oriented decision modelling method, such as extending its evaluation on real decision-making scenarios in software evolution and dynamic adaptation taken from industrial use cases of the SUPERSEDE project. In parallel, we plan to further explore possible synergies with decision taxonomies, such as GRADE, and other goal-oriented languages, such as GRL.

5 References

1. Malpas, J., "Donald Davidson", The Stanford Encyclopedia of Philosophy (Winter 2012 Edition), Edward N. Zalta (ed.), http://plato.stanford.edu/archives/win2012/entries/davidson/.
2. OMG. Decision Model and Notation (DMN) v 1.1. (2016) Available online at: https://www.omg.org/spec/DMN/About-DMN/, last accessed 2018/02/28.
3. Janssens, L., Bazhenova, E., De Smedt, J., Vanthienen, J., Denecker, M.: Consistent Integration of Decision (DMN) and Process (BPMN) Models. In: España, S., Ivanović, M., Savić, M. (eds.) CAiSE Forum 2016, 121-128. CEUR Workshop Proceedings (2016).
4. Guizzardi, R.S.S., Perini A., Susi A.: Aligning Goal and Decision Modeling, In Proceedings of CAiSE Forum, 2018, To appear.
5. Giorgini, P., Maiden, N., Mylopoulos, J. and Yu, E. (eds.) Social Modeling for Requirements Engineering, MIT Press (2010).
6. van Lamsweerde, A.: Reasoning About Alternative Requirements Options. Conceptual Modeling: Foundations and Applications 2009: 380-397
7. Jiang, L., Barone, D., Amyot, D., Mylopoulos, J.: Strategic Models for Business Intelligence. In: ER 2011, 429-439. Springer, Heidelberg (2011).
8. E. Papatheocharous, K. Petersen, A. Cicchetti, S. Sentilles, S. M. A. Shah, T. Gorschek, Decision support for choosing architectural assets in the development of software-intensive systems: The GRADE taxonomy, in: Proceedings of the 2015 European Conference on Software Architecture Workshops, ACM, 48, 2015.
9. Taylor, J., Debevoise, T.: The MicroGuide to Process and Decision Modeling in BPMN/DMN: Building More Effective Processes by Integrating Process Modeling with Decision Modeling, ISBN 978-1-502-78964-8, 2014.