Trying to Elicit and Assign Goals to the Right Actors

Anouck Chan, Anthony Fernandes Pires, Thomas Polacsek

Trying to Elicit and Assign Goals to the Right Actors

Anouck Chan(✉), Anthony Fernandes Pires(✉), and Thomas Polacsek(✉)

ONERA, Toulouse, France
{anouck.chan,anthony.fernandes_pires,thomas.polacsek}@onera.fr

Abstract. At the beginning of a project, an organisation may define very abstract goals. These high-level goals describe organisation characteristics that all projects must fulfil. Due to the very generic and abstract nature of these goals, it is sometimes not easy to break them down into more concrete goals and to decide who should be responsible for what. For many years, goal modelling approaches have proposed frameworks for eliciting and defining stakeholders' goals in an organisation. In the context of an aeronautical company, we conducted an application on a case of study of a goal modelling method. From high-level goals, we have supported business experts in eliciting more concrete goals, assigning them to the right actors and identifying possible organisational needs. For this, we started from an existing method that we have adapted to fit our purposes.

Keywords: Goal modeling · Requirements engineering

1 Introduction

Goals are widely studied in the context of Goal-Oriented Requirements Engineering (GORE) [8,10,14]. Unlike a requirement, a goal is not mandatory, it is more a prescriptive statement. Goals express the objectives that the system should achieve [9]. Interactions between goals and interactions among goals and other elements like actors or resources are finely studied in GORE frameworks such as *Non-Functional Requirements* (NFR) [3,13], iStar [4,16] or KAOS [5].

Goals are generally quite concrete objectives whose responsibilities are clearly defined, even if they are used in the preliminary phases of projects. Knowing which goal is under whose responsibility is a crucial organisational element. In practice, a growing number of organisations are incorporating very abstract goals, namely *high-level goals*, whose assignment and translation into requirements are not trivial. Moreover, these high-level goals are often derived from the activity the company wants to conduct and the value the company is driven by and wants to convey in its products. Take the basic example of a watch manufacturer, whose activity is to create watches and who may choose to reflect the value of modernity or, on the contrary, of tradition. In one case, the creation of the watches will be driven by goals to show that they are at the cutting edge of innovation, while in the other the emphasis will be on historical continuity or a heritage of craftsmanship.

We were approached by an industrial firm, an aircraft manufacturer, who has developed a value repository. These values are not only applied to aircraft, but also to the company itself, whether it is on manufacturing, operational services or maintenance operations. The notion of value used here is the same as the one used in psychology, as it is generally the case for companies [7,12]: values are "*concepts or beliefs, about desirable end states or behaviors, that transcend specific situations, guide selection or evaluation of behavior and events, and are ordered by relative importance*" [15]. When the aircraft manufacturer contacted us, high-level goals were already derived from the company values and its activity[1]. The problem for the company was how to elicit more concrete goals from these high-level goals and assigning these concrete goals to the right actor who has the skills to ensure their satisfaction.

The contribution of this article is to present an adaptation of an existing GORE methodology to address a real industrial problem. As presented in Sect. 2, we have searched through the research literature an existing methodology using refinement and delegation through actors to obtain a satisfying set of goals from a global objective. Then, we introduce an adaptation of an existing method to suit our problem in Sect. 3. Section 4 is dedicated to the application of our approach to three aircraft manufacturer high-level goals. Based on this, we draw some observations and conclude in Sect. 5.

2 Looking for a Methodology

Several works address the problem of goal elicitation and goal refinement. However, very few focus on the refinement of very abstract goals into more concrete goals assigned to actors.

The refinement calculus CaRE proposes a tool to build a consistent, complete and realisable set of requirements (specifications) from an initial and incomplete one given by stakeholders [6]. Each requirement can be discarded or modified as long as there is an acceptable chain of arguments for this. The chain consists in an alternation between defects of the requirements and proposals for their resolution in the form of requirements refinement named operators. This method is mainly oriented to obtain a coherent set of requirements, avoiding conflicting or inconsistent ones. Finding a solution to satisfy the initial set of goals is therefore not a priority, nor defining a set of actors who can satisfy the refined goals. This is why the method is not completely suited to our study situation.

The method presented in [11] allows to elicit satisfiable subgoals from an initial goal and to distribute them to agents able to satisfy them. Goals are expressed with modal logic and each agent can control and monitor some variables. An agent can satisfy a goal only if it has control on the goal variables. Because some goals may not be realisable, the authors present *tactics* (*i.e.*

[1] How high-level goals are derived from values and activities is a point that deserves much investigation, but this is out of the scope of this article.

schemes) for refining them into satisfiable subgoals. These tactics consist in a guideline to resolve the issue for each condition for which a goal is unrealisable. In our context, the main restriction for applying this method is the fact that goals are expressed with variables and temporal operators. The high-level goals we use cannot be described in such details, either because they are too abstract or because their description would involve too many variables. Moreover, we are not interested in how each goal is satisfied but only in who is responsible for its satisfaction.

Bryl, Giorgini and Mylopoulos present a method to find an optimal set of actors' actions that satisfy an initial set of goals [1,2]. They use predicate logic to describe goals, actors' properties, and relations between actors. The actors have some autonomy and personal objectives. As the defection of one actor could be prejudicial to the achievement of the initial objectives, the method tries to ensure actors' collaboration by computing *stable* solutions.

Each actor is described through its capacities to execute actions and their assumption about the other actors' capacities.

A special actor, namely the *manager*, starts the process. It refines high-level objectives into subgoals and delegates them to other actors. Then, steps are iterated until a *stop-criterion* is reached (*e.g.* time limit). At each step, actors explore alternatives to achieve all goals that have been assigned to them and select the best alternative according to their criteria. For each goal, an actor can choose between three actions: *satisfy* it, *refine* it into subgoals or *delegate* it to another actor. Refinement can be *AND*-type, to divide goals in simpler ones, or *OR*-type, to elicit then choose the favourite alternative. Delegating allows an actor who has a goal it cannot, or decide to not, satisfy itself, to give it to another actor. A stable solution is reached when no actor is willing or able to change any of its actions.

3 Our Approach

Several elements of Bryl *et al.* approach are close to our problem while others need adjustments.

Firstly, the combination of refining and delegating actions are a way for actors to work together to achieve a global goal and to ensure their collaboration. The whole process is based on the knowledge of the actors on how to meet a goal. This is particularly relevant in situations where the actors have specific and dedicated know-how as ours. Therefore, we keep the refining and delegating actions. Actors personal objectives are out of scope of our problem, *i.e.* we only consider the company's high-level goals satisfaction. Thus, we are no longer looking for a stable solution from which the actors do not want to deviate, but a solution in which each goal is assigned to an actor that can satisfy it.

Secondly, goals refinement can be stopped as soon as an actor is able to satisfy the goal. Thus, actors may have a great autonomy on how achieve their goals without harming the global objectives. However, the solution obtained is dependent on actors initial description, in particular of how they can refine and to whom they can delegate, which makes the first method step, *i.e.* the manager step, quite crucial. In that respect, the method could end in a final situation where an actor can neither satisfy a goal assigned to it, nor delegate it to another actor. In order to avoid this issue, we add a set of unsatisfiable goals named *Unknown*. As detailed later, this set is made up of goals that the company does not yet know how to satisfy or to assign to.

Finally, in Bryl *et al.*, actors choose the actions they perform. We focus on elicitation and not a multi-agent decision elicitation system. So, we use a common pool of knowledge (who can satisfy what) and allow all possible delegations. A central entity decides what actions are performed. This is made possible by the fact that no actor wants to deviate from the solution and by the addition of the *Unknown* set that models all potential new ways to satisfy unsatisfiable goals.

Key Concepts. The key concepts of our adaptation can be described as follows. *High-level goals* are the main input of our approach. They correspond to abstract objectives, whose achievement is not obvious. High-level goals are linked to a *value* and an *activity*, which are out of the scope of our method. Actors we consider can be a person or a group of persons (seen as a single entity). A *goal* is an objective that should be satisfied and can be assigned to an actor. In the first step of the method, a goal is derived from each high-level goal. All actors have *skills* and skills are required to satisfy goals. If an actor possesses all the skills required to satisfy a goal assigned to it, the latter is a *satisfiable goal.* If no actor possesses -even partially- the required skills to satisfy the goal, it is an *unsatisfiable goal.* Otherwise, the goal can be *refined* into subgoals or can be *delegated* to another actor. A goal is satisfied if it is a satisfiable goal or if all its subgoals are satisfied. The concept of *Unknown* set corresponds to a set where unsatisfiable goals are stored. Therefore, all *unsatisfiable goals* are *de facto* delegated to the *Unknown set.*

Method. The inputs of our algorithm are the company's high-level goals HG and the set of actors $\mathcal{A}$. The set of goals handled by the approach is denoted $\mathcal{G}$ and the set of skills required for satisfying a goal $g \in \mathcal{G}$ is denoted $\mathit{skills}(g)$. Note that set $\mathcal{G}$ is built iteratively by the algorithm. For each actor $a \in \mathcal{A}$, $\mathit{skills}(a)$ denotes the set of skills of a. We also consider three functions, *satisfy*, *delegate* and *refine*, that correspond to actions that can be performed by actors. The result of applying the *refine* function on an actor $a \in \mathcal{A}$ and a goal $g \in \mathcal{G}$ is a pair of goals (g', g'') such that: 1. the satisfaction of subgoals g' and g'' implies the satisfaction g, 2. g' is satisfiable by a ($\mathit{skills}(g') = \mathit{skills}(g) \cap \mathit{skills}(a)$) and 3. g'' is not satisfiable by a, even partially ($\mathit{skills}(g'') = \mathit{skills}(g) \setminus \mathit{skills}(a)$).

The output of our algorithm is, for each actor $a \in \mathcal{A}$, the set of its satisfiable goals $\mathcal{G}^a$ and the *Unknown* set of unsatisfiable goals $\mathcal{U}$.

At the beginning of our algorithm, $\mathcal{U}$ is empty. Each high-level goal in HG is refined into a goal g that is assigned to an actor a such that a can, even partially, satisfy it ($\mathit{skills}(a) \cap \mathit{skills}(g) \neq \emptyset$). We assume that this initialisation is always possible.

Our algorithm is then a loop that ends when all goals are either satisfied or assigned to $\mathcal{U}$. As long as there exists an actor a with a goal g assigned to it and that is not satisfied yet, if $\mathit{skills}(g) \subseteq \mathit{skills}(a)$ then g is satisfied and added to the set of satisfiable goals of a ($\mathcal{G}^a$ becomes $\{g\} \cup \mathcal{G}^a$). Otherwise, the function *refine* is performed to obtain the subgoals g' and g'' as described earlier, and g' is satisfied and added to $\mathcal{G}^a$. Regarding g'', if there exists an actor a' in $\mathcal{A}$ that can, even partially satisfy it ($\exists a' \in \mathcal{A}, \mathit{skills}(a') \cap \mathit{skills}(g'') \neq \emptyset$) then g'' is assigned to a' else g' is added to $\mathcal{U}$.

4 Application to Our Industrial Problem

In this section we present the application of our method in an industrial context and give the lessons learned from our case of study application.

4.1 Modus Operandi

We have worked with an aircraft manufacturer who has developed a value repository, along with their associated high-level goals. During the whole case study application, we have worked with three company high-level goals.

First, we have the high-level goal *Aircraft Deliverability* derived from the value *Deliverability* and the activity *Producing the aircraft*. The value *Deliverability* means the capacity of the system to deliver as expected. So this high-level goal is the ability to deliver the aircraft to customers when they are needed at the desired rate. Second, we have the high-level goal *Industrial System Deliverability* derived from the value *Deliverability* and the activity *Producing the industrial system*. This high-level goal is the ability to produce the factories, assembly lines and infrastructures needed to manufacture the aircraft. Third, we have the high-level goal *Industrial System Performance* derived from the value *Performance* and the activity *Producing the industrial system*. The value *Performance* means the capacity of the system to work as expected. So this high-level goal is the ability of the industrial system to perform according to production criteria.

The application was conducted with two aircraft manufacturer experts. The first one is specialised in the architecture aircraft design, the second in methods and digital solutions for design and manufacturing. Both of them have been involved for many years in company projects aiming at linking the design of the aircraft with the design of the means of production. Prior to our work, the two experts had some notions in goal modelling, especially concerning the notion of actors and goals, but had never used any refinement methods.

We have first organised a three-hours long session dedicated to our method presentation. To do so, we worked on a simple example of high-level goals refinement and actor assignment. Then, for each high-level goal, we have worked with the experts during two three-hour sessions. The objective of the first session was to apply the method. At the beginning of the session, we identified a set of potential actors and then applied our algorithm. As there were some actors that we had not thought of until we reached a certain stage in the process, we repeated algorithm application several times. At the end of the session, a first model was obtained. The second session took place a week later and its purpose was to amend and improve the model collectively. Between the two sessions, we reviewed the model and identified issues to be fixed with the experts. An issue could be, for example, that we did not understand the description of a goal, or that a goal delegated once to an actor, was delegated again to the initial actor without being refined.

Finally, two months later, three one hour long sessions allowed us to finalise the model by working on specific issues like language issues or actor description.

4.2 Goal Elicitation and Assignation

Figure 1 describes the application of our method to the high-level goals *Aircraft Deliverability* and *Industrial System Deliverability*. These two high-level goals become respectively the goals D1 and D2 and were assigned by the experts to two actors: the *Industrial System Operator*, in charge of operating the industrial system to produce aircrafts on a daily basis, and the *Industrial System Developer*, in charge of the design of the industrial system and its development.

The *Industrial System Operator* cannot completely satisfy the goal D1 , so the goal is refined into two goals: one related to the supply chain (goal D1a), the other related to the line balancing (goal D1b). Because this actor is in charge of logistics, the D1a goal is her responsibility. Goal D1b, on the other hand, is not her competence. Line balancing consists of scheduling the assembly tasks in order to obtain the best production rate. D1b is therefore not a goal that depends on the operational aspect, but on the design of the assembly line. So D1b is delegated to the *Assembly Line Developer* actor. D1b is in turn refined into four goals. Two are not in the actor competence scope: D1bc which is a goal for the daily operations and D1ba which aims at setting the assembly tasks. Goal D1ba is the responsibility of a new actor the *Aircraft Architect*, the actor who is in charge of the aircraft design. Indeed, the assembly actions depend directly on the design choices made by the *Aircraft Architect* and therefore they can influence these choices depending on the complexity of the tasks to be performed. However, the tasks at the architectural level are too abstract, they need to be refined to a more basic level, the actions performed by the workers. Therefore, D1ba cannot be completely satisfied by the *Aircraft Architect*, so the goal is refined into two goals: one is to provide the aircraft design (goal D1baa), which implies defining the assembly tasks at the architectural level, and another one is to define the detailed assembly tasks (goal D1bab). Because it belongs to the work of the line workers, this latter goal is assigned to the *Industrial System Operator* actor.

Now let's look at the goal D2. Through the refinements of this goal, we see three goals that belong to the actor, D2cb, D2ab and D2ac, and two goals that are transferred to the *Assembly Line Developer* actor. In fact, goals D2cb, D2ab and D2ac deal with the ability to build and evolve infrastructures such as buildings, roads, etc., while goals D2ca and D2b deal with assembly line tools and their possible evolution. Goal *Design the aircraft parts transport infrastructure* (goal

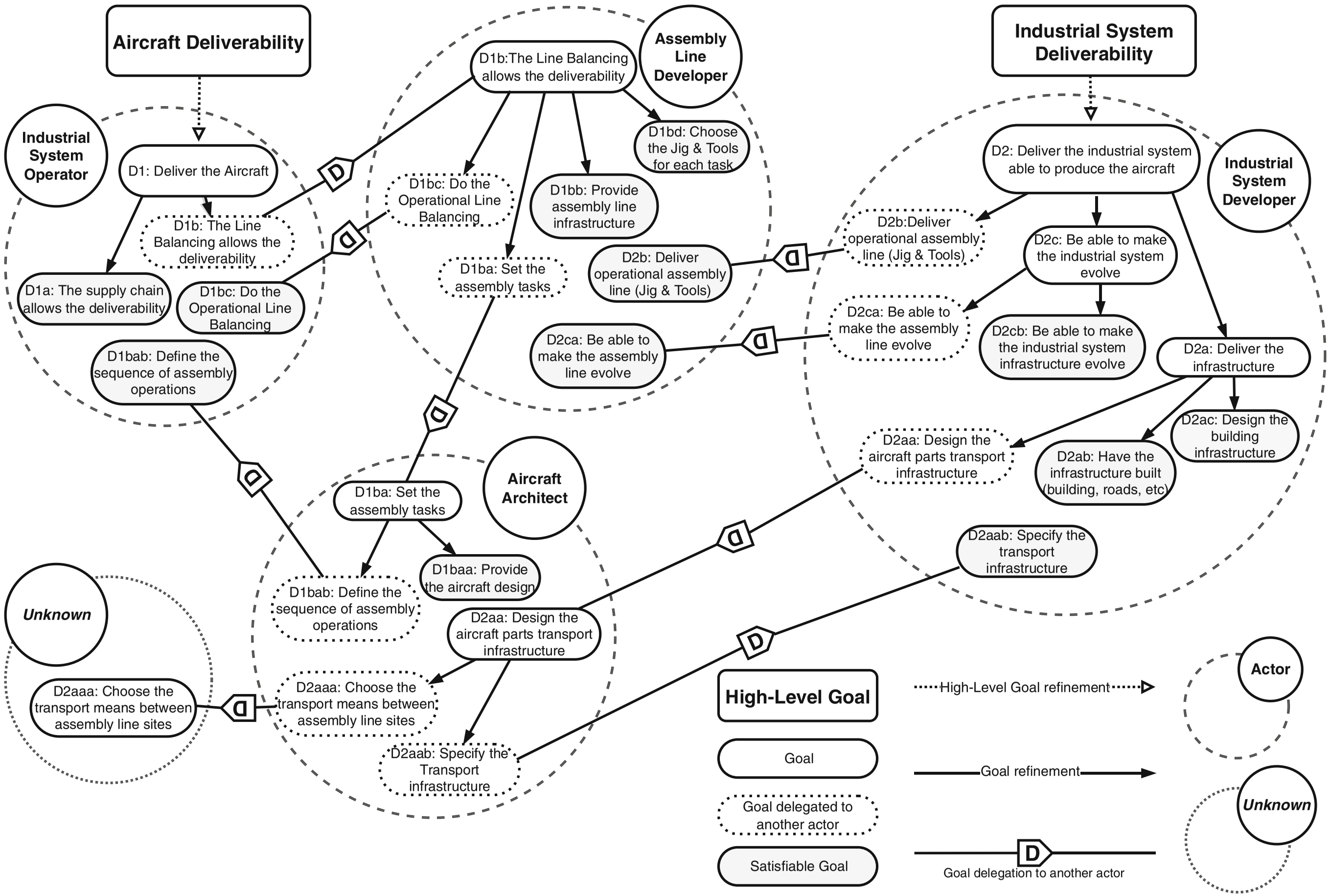

Fig. 1. Application for the high-level goals *Aircraft Deliverability* and *Industrial System Deliverability*

D2aa) is particularly interesting. The means of transport, as well as their size and therefore routes they take (air, land or sea), depend greatly on the aircraft elements to be conveyed. We find this dependency on the D2aa decomposition, where the goal is first delegated to the *Aircraft Architect*, since the aircraft design is her responsibility, and then decomposed into two goals. The first, D2aaa, consists of choosing the transport means. The second one, D2aab, is for the actor in charge of the industrial system design. Although D2aab can be achieved by the *Industrial System Developer*, D2aaa can only be satisfied by a combination of interrelated skills from the *Aircraft Architect* and another actor. As the experts couldn't define a clear-cut between the part of this goal satisfiable by the *Aircraft Architect* and the one not satisfiable, D2aaa is delegated to the *Unknown* set.

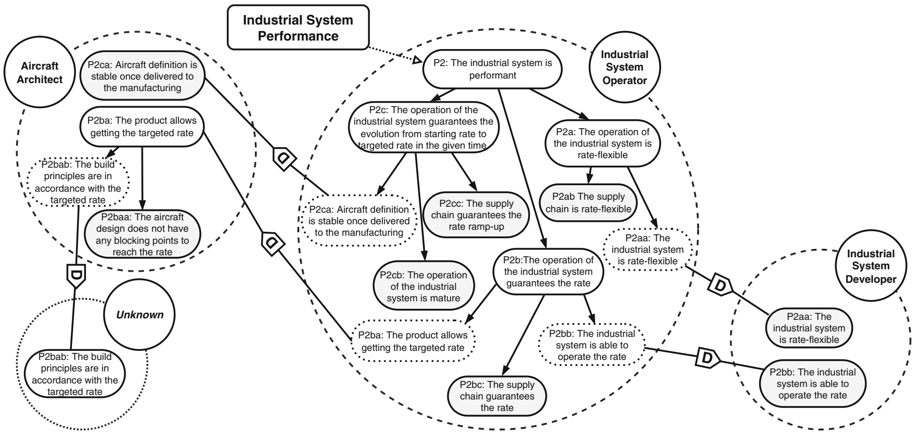

Fig. 2. Application for the high-level goal *Industrial System Performance*

In the same way, we applied our method to the high-level goal *Industrial System Performance*. The result is illustrated in Fig. 2. The high-level goal is refined into the goal P2 and assigned to the actor *Industrial System Operator*. In this example too, we can notice the presence of a goal that is not assigned to any actor: P2bab. This goal is related to the production rate and more specifically to the fact that the aircraft design should not be an obstacle to its manufacture. By using our method, we see that the performance of the industrial system involves, among other things, the ability of the system to hold the desired rate. This implies that the aircraft is designed to achieve the rate. An example of a bottleneck would be the use of technology, like a special alloy, that is too time consuming on the assembly line. This goal is not the responsibility of the *Industrial System Operator*, but rather of the *Aircraft Architect*. The *Aircraft Architect* is in turn blocked because solving this goal requires mixed knowledge, from aircraft design and production. So the goal P2bab is assigned to the *Unknown* set.

5 Conclusion

Following this work, we have discussed our results with the two experts and with a third architect specialist in product/production relationships. The methodology was well received. They particularly appreciated having a precise and reproducible method to clearly set everyone's goals, and the ability to bring out the need to create new actors to manage the goals derived from the high-level goals. From this first experiment, we can draw some observation.

Firstly, in our algorithm, the refinement of one goal leads to two goals, not more. In our use case, we frequently violated this rule. It is not clear whether this deviation comes from our method being too restrictive or whether we skipped intermediate steps in the refinement during the experiment. Secondly, sometimes we had an abundance of information and it was difficult to capture all of it. We sometimes got lost in the rich expert knowledge: they know so many details about their domains that we had a hard time going back to an abstract level. Thirdly, it was also difficult to express the goals in terms of wording directly from the first session. Even if the experts had a good vision of what needed to be done to achieve a goal, it was often in the form of actions or process (i.e. sequences of actions) which was very difficult to formulate as goals. A rewording of the goals often happened in the second session. Finally, some goals are indeed delegated to the *Unknown* set. In this use case, they are nobody's responsibility because the experts could not decide which actor could satisfy them. Practically speaking, such goals require an indivisible combination of different actors skills, who therefore have to actively cooperate. This raises the question of creating within the company these new actors able to act as a bridge among the different actors.

Regarding future work, we plan to investigate further the preliminary lessons learned in order to improve the method. Then, we need to extend it in order to give solutions to the user for the goals delegated to the *Unknown* set. It could be a way to identify missing important actors in the company or to define the need for strong cooperation between existing actors. We also need to examine goals of the actors which are not elicited through the method but still important for the actors and the company. We will have to understand the meaning of their absence from any high-level goal refinements for the company. In addition, we would like to join this method with traditional GORE approaches. The idea would be to use the resulting elicitation of goals and actors as an input for further goals refinement and dependencies elicitation. Finally, we could also investigate the derivation of high-level goals from values, in order to procure traceability between company values and goals assigned to actors.

Acknowledgements. We want to warmly thank François Bouissière, Claude Cuiller and Pierre-Éric Dereux, who allowed us to work on this case study, and Stéphanie Roussel for her support. Their encouragement, their availability and their comments were invaluable in the development of this work.

References

1. Bryl, V., Giorgini, P., Mylopoulos, J.: Designing cooperative IS: exploring and evaluating alternatives. In: Meersman, R., Tari, Z. (eds.) OTM 2006. LNCS, vol. 4275, pp. 533–550. Springer, Heidelberg (2006). https://doi.org/10.1007/11914853_32
2. Bryl, V., Giorgini, P., Mylopoulos, J.: Designing socio-technical systems: from stakeholder goals to social networks. Requir. Eng. **14**, 47–70 (2008)
3. Chung, L., Nixon, B.A., Yu, E., Mylopoulos, J.: Non-Functional Requirements in Software Engineering. International Series in Software Engineering, vol. 5. Springer, New York (2000). https://doi.org/10.1007/978-1-4615-5269-7
4. Dalpiaz, F., Franch, X., Horkoff, J.: iStar 2.0 language guide. CoRR abs/1605.07767 (2016)
5. Dardenne, A., van Lamsweerde, A., Fickas, S.: Goal-directed requirements acquisition. Sci. Comput. Progr. **20**(1–2), 3–50 (1993)
6. ElRakaiby, Y., Borgida, A., Ferrari, A., Mylopoulos, J.: A refinement calculus for requirements engineering based on argumentation theory. In: Dobbie, G., Frank, U., Kappel, G., Liddle, S.W., Mayr, H.C. (eds.) ER 2020. LNCS, vol. 12400, pp. 3–18. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-62522-1_1
7. Ferrario, M.A., Simm, W., Forshaw, S., Gradinar, A., Smith, M.T., Smith, I.C.: Values-first SE: research principles in practice. In: IEEE/ACM 38th International Conference on Software Engineering Companion (ICSE-C), pp. 553–562. ACM (2016)
8. Horkoff, J., et al.: Goal-oriented requirements engineering: a systematic literature map. In: 24th IEEE International Requirements Engineering Conference (RE), pp. 106–115. IEEE (2016)
9. van Lamsweerde, A.: Requirements engineering in the year 00: a research perspective. In: Proceedings of the 22nd International Conference on on Software Engineering, ICSE, pp. 5–19. ACM (2000)
10. van Lamsweerde, A.: Goal-oriented requirements engineering: a guided tour. In: 5th IEEE International Symposium on Requirements Engineering (RE), p. 249. IEEE (2001)
11. Letier, E., van Lamsweerde, A.: Agent-based tactics for goal-oriented requirements elaboration. In: Proceedings of the 24th International Conference on Software Engineering (ICSE), pp. 83–93 (2002)
12. Mougouei, D., Perera, H., Hussain, W., Shams, R.A., Whittle, J.: Operationalizing human values in software: a research roadmap. In: Proceedings of the ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, ESEC/SIGSOFT FSE, pp. 780–784. ACM (2018)
13. Mylopoulos, J., Chung, L., Nixon, B.A.: Representing and using nonfunctional requirements: a process-oriented approach. IEEE Trans. Softw. Eng. **18**(6), 483–497 (1992)
14. Mylopoulos, J., Chung, L., Yu, E.S.K.: From object-oriented to goal-oriented requirements analysis. Commun. ACM **42**(1), 31–37 (1999)
15. Schwartz, S.H., Bilsky, W.: Toward a psychological structure of human values. J. Personal. Soc. Psychol. **53**, 550–562 (1987)
16. Yu, E.S.: Towards modelling and reasoning support for early-phase requirements engineering. In: Proceedings of the 3rd IEEE International Symposium on Requirements Engineering (ISRE), pp. 226–235. IEEE (1997)