On Temporally Annotating Goal Models

Sotirios Liaskos[1] and John Mylopoulos[2]

[1] School of Information Technology, York University
Toronto, Canada
liaskos@yorku.ca

[2] Department of Computer Science, University of Toronto
Toronto, Canada
jm@cs.toronto.edu

Abstract. Goal models are theories that describe how various stakeholder goals relate to each other. The constructs that such models use to represent these relationships focus on characterizing the nature of causality that connects goals, without, however, including temporal aspects such as the order in which goal satisfaction takes place. Nevertheless, introducing constructs to allow explicit representation of this ordering aspect has been shown to be useful for a variety of applications. Furthermore, representation of such information need not necessarily be done through formalization or use of external representations; it is also possible through simple annotations on the core goal model. This allows for representing the temporal dimension of goal models in a lightweight and concise manner. However, it does not come without influencing the established way to perceive goal models. In this paper, we discuss our experience in augmenting goal models with temporal information about goal satisfaction, which we performed for the purpose of representing and reasoning about behavioral variability.

Keywords: requirements engineering, goal modeling, i-star

1 Introduction

Goal models constitute theories of how stakeholder goals relate to each other, through the representation of a variety of relationships amongst them, such as means-ends, decomposition and contribution links. In their semi-formal form, i.e. in *i**, goal models represent these relationships in a time-independent way, leaving temporal and ordering considerations outside their scope. This has not prevented these models to prove remarkably useful tools for understanding and reasoning about domains and early requirements, through offering a clean representation of the intentional aspect of a requirements problem.

However, explicit representation of ordering and temporal aspects of goal fulfillment, particularly in the form of direct annotations to the goal diagrams, has often been attempted in the *i**-related literature ([6, 3, 2]). In these efforts, adding ordering annotations constitutes a preparatory step for eventually translating the graphical model into a formal representation (such as Golog or Formal Tropos) for subsequent enrichment and analysis. This seems to be an indication that, for many, such annotations add value to plain graphical goal models in terms of not only guiding the subsequent formalization process, but also simply communicating this additional aspect of goal fulfillment – the temporal one.

In this short presentation we discuss our own attempts for augmenting semi-formal goal models with temporal information for the purpose of preference-based variability analysis and offer some insights on the benefits and caveats that may accompany such an effort.

2 Objectives of Research

In our recent work we have been studying the potential of using goal decomposition structures, in order to automatically reason about variability in the problem domain ([4]). The structures we are using are single-agent AND/OR decomposition models which are directly analogous to means-ends and decomposition trees that may appear in *i** strategic rationale diagrams. In these models, variability is expressed through the presence of OR-decompositions, which in *i** terms reflect alternative means (subgoals) to fulfill certain ends (parent goal). The presence of OR-decompositions implies that there are alternative *sets* of leaf level tasks that constitute solutions of the problem described by the root goal – these sets are solutions of the AND/OR-decomposition tree. Thus, by constructing such trees, significant amounts of requirements *variability* can be concisely captured. Furthermore, the presence of contribution links of each alternative to soft-goals allows us to further raise the level of abstraction in which reasoning about goal variability can take place.

Nevertheless, we found that there is significant amount of variability in ordering (temporal) aspects of goal fulfillment that is not captured this way. A merchant desires to *Send a Shipment* after the *Payment is Received* for particular customers; but not for others. A meeting initiator may *Announce a meeting* only after the *Meeting Room is Confirmed*, or she may not be so cautious, depending on the nature of the meeting and how busy the room is known to be. At a lower level, an ATM system may or may not *Provide the Card Back* to the user before *Money is Dispensed*. The existing core constructs for building semi-formal goal models do not seem to explicitly accommodate such ordering relationships between goals and tasks.

In our work, we explore ways by which this type of variability can be represented in goal models and subsequently translated into forms that allow useful reasoning about alternatives. The literature seems to offer two fundamental alternatives for adding a temporal dimension to goal models. One is exhaustive formalization of individual goals using for example LTL (see KAOS - [1]), a practice that, although useful for rigorous analysis, it could be characterized as requiring significant effort investment and expertise, while offering a result of potentially reduced comprehensibility. A second alternative, which maintains low formality, is the use of an external representation in sync with the goal models as done in [5] where use case maps are used. However, external representations may also require a minimum of additional non intention-related information to be defined. We believe that direct annotation on the goal models with ordering information, is also a useful practical alternative and, if done carefully, it can significantly increase the amount of information that goal models convey about the domain. We sketch how we do it in our work in the next section.

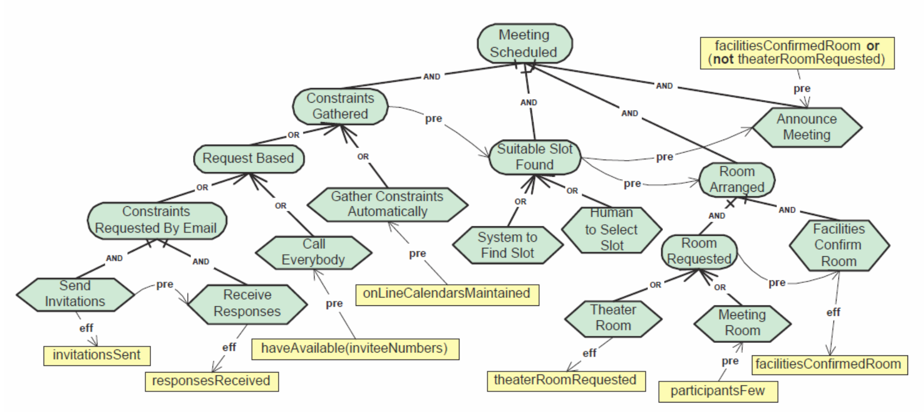

Fig. 1. Goal Models with Temporal Annotations

3 Contributions: Modeling Ordering Constraints

In our proposal for temporally extending goal models, we simply considered that the ordering of goal fulfillment and task performance is relevant. Thus what potentially fulfills the top level goal is not just a set of tasks but a *sequence* thereof. We call such sequences *plans*, borrowing from the corresponding term used in the AI-planning community. A consequence of this extension is that the number of alternatives to be analyzed increases dramatically: the number of solutions of the AND/OR structure is now multiplied by the possible orderings of the tasks that constitute each solution.

Visual representation of ordering constraints on the model itself serves the purpose of controlling the comprehensibility issue that this explosion introduces to users of the model. We introduced the *precedence link* that connects hard elements (hard-goals or tasks) with each other. A precedence link between two goals or tasks implies that satisfaction (resp. performance) of the latter is not possible unless the former has been satisfied (resp. performed) first. In Figure 1, precedence links applied to a goal model for the Meeting Scheduling example are shown. Thus, the task *Facilities Confirm Room* can be performed only if the goal *Room Requested* has been satisfied, hence the precedence link. As seen in the figure, in our application, indicative temporal constraints dictated by the domain realities rather the stakeholder desires, were found to be served very well by the precedence link.

A somewhat bolder addition to the graphical representation that we have attempted is that of domain variables playing the role of precondition and effects of tasks. Domain variables, in the form of predicates, are used to represent the *state* of the environment in which performance of tasks and achievement of goals is attempted. These can describe both volatile and more stable facts in the domain. Thus, in meeting scheduling, *invitationSent*, *haveAvailable(inviteeNumbers)*, *participantsFew* or *theaterRoomRequested* are examples of domain variables. Simple logical expressions of these variables can then be constructed and set as precedence conditions inside the diagram itself as seen in Figure 1. For example, the expression linked to the task *Announce Meeting* on the right end of the figure says that this task should be performed only after room confirmation by the facilities office has been performed unless it was not the theater room that was requested. In addition, such predicates or lists thereof can be set as *effects* of the performance of task. For example, when the task *Send Invitations* is performed the domain variable *invitationsSent* becomes true. Note that upon introduction of domain variables, plans need to be calculated subject to predefined initial conditions. Through the addition of environmental variables inside precedence and effect elements, plain goal models can be used to visually construct a skeleton of a dynamic domain, to be subsequently automatically translated in a more formal form for further enrichment and analysis.

As we describe below, these seemingly small additions introduce interesting possibilities but also influence the way that a goal model is read and understood, while introducing some challenging conceptual issues.

4 Conclusions

We have applied the visual annotations we described in a variety of goal models either based on artificial data (e.g. Meeting Scheduler or ATM) or based on projects that offered more realistic data (e.g. on the nursing domain). While a formal empirical study is still pending, our so far exploration does not offer evidence that viewing goal models from a temporal point of view obstructs comprehension of the model or the domain. We have however realized that certain implications must be made explicit to avoid ambiguity and lift potential perception problems.

As example of the kind of attention that needs to be paid, consider the precedence link. While the link implies that the destination must be preceded by the origin, it does not necessitate that the origin must be followed by the destination; the latter may as well be absent. To represent that the destination must follow the origin, whenever the latter is satisfied or performed, a different link definition is needed, e.g. a response link. In such a link, non-satisfaction of the origin does not prevent satisfaction of the destination. Note that these two definitions have different consequences when the origin is alternative or optional.

As another example of how temporally extended goal models influence the established understanding of goal models, consider the relationship between soft-goals and precedence. Firstly, direct definition of plain precedence between soft-goals is not necessarily intuitive. For example, simply specifying that *Privacy* is satisfied before *Convenience* may be an unintuitive statement, when for both soft-goals crisp satisfaction criteria do not exist in the first place. Indirect precedence in soft-goal satisfaction via *hurts* and *helps* contributions from ordered hard elements, however, is possible. But this leads us to an understanding of soft-goal satisfaction as a degree (e.g. of satisfaction evidence) that fluctuates during the course of a plan. Reasoning about soft-goal satisfaction is then necessarily subjected to temporal modalities. Thus statements of the form “we want Privacy to be at least partially satisfied” are now extended as “we *never/always/eventually* want Privacy to *never/always/eventually* be at least partially satisfied”.

We found that the modeler who wishes to rely on visual annotations of temporal constraints needs to address issues such as the above by cleanly defining the intuitive meaning of each introduced construct and explaining its implications. This is separate and in addition to the definition of formal semantics, which does not necessarily support comprehension when users are not trained to the underlying formalism (PDDL, Situation Calculus, Formal Tropos/SMV, LTL etc).

5 On-going and Future Work

Our current understanding is that, despite its potential cost and time investment, the best tool to validate compliance with comprehensibility and usability criteria we set for our extension proposals is the empirical study. In our work, we aim at developing variability and preference representation and analysis techniques that are usable not only by analysts whose training may not include understanding of formal languages, but potentially also by common software users with no experience in any kind of conceptual modeling whatsoever. Empirical investigation implies experimental designs that may include a variety of tests, such as successful communication of the intended meaning of the model, the effort it takes for this to happen or the effort it takes for the construction of temporally annotated models and how correctly this is done for different participant profiles in terms of their experience in modeling. These plans for empirical work have gradually claimed a larger and larger piece of our research agenda.

References

1. Anne Dardenne, Axel van Lamsweerde, and Stephen Fickas. Goal-directed requirements acquisition. *Science of Computer Programming*, 20(1-2):3–50, 1993.
2. Ariel Fuxman, Lin Liu, John Mylopoulos, Marco Pistore, Marco Roveri, and Paolo Traverso. Specifying and analyzing early requirements in Tropos. *Requirements Engineering*, 9(2):132–150, 2004.
3. G. Gans, M. Jarke, G. Lakemeyer, and T. Vits. Snet: A modeling and simulation environment for agent networks based on i* and ConGolog. In *Proceedings of the 14th International Conference on Advanced Information Systems Engineering (CAiSE'02)*, Toronto, Canada, 2002.
4. Sotirios Liaskos, Sheila McIlraith, and John Mylopoulos. Towards augmenting requirements models with preferences. In *Proceedings of the 24th IEEE/ACM International Conference on Automated Software Engineering (ASE 2009)*, Auckland, New Zealand, 2009.
5. Lin Liu and Eric Yu. Designing web-based systems in social context: A goal and scenario based approach. In *Proceedings of the 14th International Conference on Advanced Information Systems Engineering (CAiSE 2002)*, Toronto, Canada, 2002.
6. X. Wang and Y. Lesperance. Agent-oriented requirements engineering using ConGolog and i*. In *AOIS-2001 Bi-Conference Workshop at Agents 2001 and CAiSE'01.*, 2001.