Tool support for modeling and reasoning with decision theoretic goal models

Sotirios Liaskos[1]

[1]School of Information Technology, York University, Toronto, Canada

Abstract

Goal models are known to be effective in capturing large numbers of alternative ways by which high-level stakeholder goals can be satisfied. Goal modeling languages such as those of the iStar family offer constructs for developing such models, and a number of software tools have been developed for supporting the modeling and alternatives analysis process. Recently, extensions of the standard iStar notation have been proposed that allow modeling of ordering constraints in goal fulfillment and task performance and of probabilistic effects of tasks. These extensions can be useful for identifying alternatives in the form of operational designs that are optimal under given risk and uncertainty assumptions. We propose a toolset for supporting modeling and subsequent analysis of such temporally and decision-theoretically (i.e., involving probabilities and utilities) extended goal models. An open-source editor is utilized for diagramming using a specially constructed shape library. A conversion tool then translates the diagrams into specifications under DT-Golog, a formal language for representing and reasoning with action theories using decision-theoretic terms. The result allows both identification of optimal policies using the DT-Golog interpreter and the answering of queries and performance of simulations using custom tools. The tool can assist in a variety of analysis tasks, ranging from modeling high-variability system behaviors and business processes to model-driven analysis of reinforcement learning domains.

Keywords

Goal Modeling, Goal Modeling Tools, Automated Reasoning, gReason

1. Introduction and significance

Goal models allow capturing large numbers of alternative ways by which high-level stakeholder goals can be analyzed into actor tasks. Modeling languages of the iStar family [1, 2] offer constructs, such as refinement and contribution links, for developing such representations. Several tools for supporting goal model development and analysis have been introduced, e.g., [3, 4, 5] – see [6] for a full list and systematic comparison. Furthermore, modeling temporal and non-deterministic aspects of goal fulfillment has also been proposed via two types of extensions to the standard iStar language [7, 8, 9]. Firstly, temporal constraints allow representation of allowable orderings by which goals can be fulfilled and tasks can be performed. Secondly, non-deterministic effects of tasks are introduced to allow modeling of alternative outcomes of task performance attempts. Rules for translating models of the thus extended language into

ER2023: Companion Proceedings of the 42nd International Conference on Conceptual Modeling: ER Forum, 7th SCME, Project Exhibitions, Posters and Demos, and Doctoral Consortium, November 06-09, 2023, Lisbon, Portugal
liaskos@yorku.ca (S. Liaskos)
https://www.yorku.ca/liaskos/ (S. Liaskos)
0000-0001-5625-5297 (S. Liaskos)

formal specifications have been proposed [9]. These allow the latter to be used for identifying optimal solutions to the goal model in the form of conditional sequences of tasks that fulfill the main operational goal while maximizing the expected satisfaction of relevant quality goals. However, a tool that can support the development and translation of such extended goal models is still absent from the gamut of goal modeling tools.

We propose *gReason*, a toolset for modeling temporally and decision-theoretically extended goal models and transforming them into specifications amenable for a variety of automated analyses. An open-source diagramming tool is used for preparing the diagrams, through a specially constructed shape library. Once completed, modelers export the diagram into its native XML format, which is then read by gReason's translation component. The latter produces a corresponding specification in DT-Golog, a formal language for decision-theoretic modeling and reasoning with action theories [10]. The DT-Golog interpreter can then be used to identify solutions of the goal model that satisfy the top level goals and also offer the highest expected reward in terms of satisfaction of related qualities. Furthermore, the specification can be used for other analyses including simulations. The latter are particularly useful for reinforcement learning (RL) tasks, whereby third-party RL components can use the simulator for training [11].

The tool implements a modeling framework that is unique in its approach to modeling non-determinism within goal models, which can, in turn, be useful for a variety of tasks, including goal-oriented business process design, the design of adaptive systems, and model-driven reinforcement learning. Through the tool, explorations of such applications, as well as case studies and other evaluation efforts become more accessible by researchers in the area.

2. Tool details

2.1. Modeling approach

An example showcasing the proposed temporal and decision-theoretic extensions of iStar can be seen in Figure 1. The standard iStar 2.0 [2] components can be found in the diagram: *goals* (ovals) are recursively decomposed into other goals or *tasks* (the hexagonal elements) using *AND-* and *OR-refinements*. *Contribution links* are added signifying how goal satisfaction and/or task performance affects satisfaction of *qualities*, as indicated by the numeric label on the link.

The extensions to the standard language are of two types. Temporal extensions show allowable ways by which goal satisfaction or task performance can be ordered. The *precedence* $\xrightarrow{pre}$ link shows that the target itself (task) or any task under it (goal) cannot be performed *unless* the origin of the link is satisfied or performed. The *negative precedence* link $\xrightarrow{npr}$, shows that the target itself (task) or any task under it (goal) cannot be performed *if* the origin of the link itself (task) or any task under it (goal) has been performed.

The second type of extensions, the decision-theoretic ones, show the effect of tasks to the state of the system under consideration. State is captured through propositional atoms whose truth status, initially false, is affected by the performance of tasks. This is shown through *effect elements* which contain one such atomic proposition. An *effect link* connects a task with a cluster of such effect elements, an *effect group*. The meaning of effect links is that performance of tasks brings about one of the effects of the effect group, via turning the truth status of the enclosed

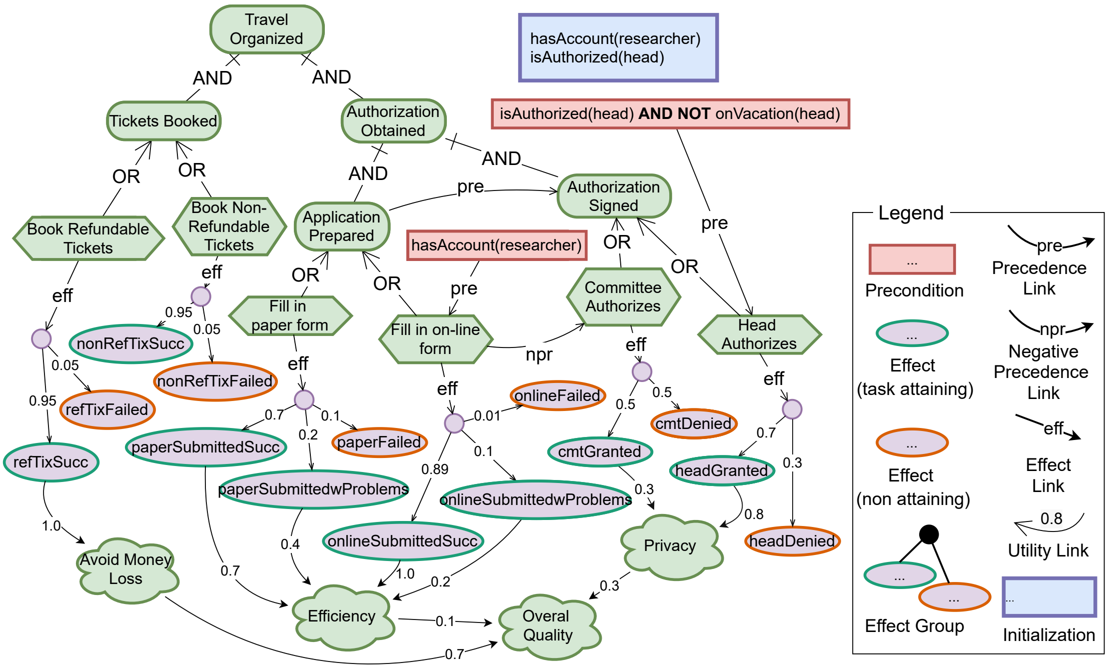

Figure 1: An example extended goal model – adapted from [9].

proposition to true. The choice of effect is probabilistic, and the value of the probability is signified by a label on the corresponding link. Of the two or more effects in a link, one or more is a *success effect* and the remaining are *failure effects*, in that occurrence of a member of the former and only the former group implies successful performance of the task. By representing successful task performance as a disjunction of the propositions contained in the success effects of its effect group, we can further represent goal satisfaction to be the formula constructed by recursively traversing the AND/OR decomposition tree under the goal, and grounded on those disjunctions. Likewise, contributions to qualities primarily originate from effects to represent that contribution to a quality from a task depends on the exact effect that the task brought about, rather than merely whether the task was attempted or not.

Finally, *precondition* boxes contain arbitrary propositional formulae of both propositions found in effects and extraneous propositions – formatted in the diagram as first-order predicates for readability – representing facts about the domain independent of task performance. The latter propositions can be initialized using an *initialization* box. Precondition elements can then be connected to goals and tasks using $\xrightarrow{pre}$ and $\xrightarrow{npr}$ links.

The resulting model can be seen as an appropriately extended *strategic rationale* (SR) view of a larger iStar diagram, representing the intentional structure of a specified actor, though explicit representation of the actor per se is currently omitted from the diagram for simplicity.

The diagramming is performed using a lightweight open-source diagramming software called *draw.io* [12]. Modelers must use a specially constructed shape library as each shape template is supplied with hidden properties informing subsequent steps of how it should be used. Draw.io allows exporting the diagram contents into its native XML format that further allows its translation to formal specifications as described next.

2.2. Translator

The main component of gReason is an application that translates the aforementioned XML export into a DT-Golog specification. The rules of the translation are complex; interested readers are referred to the latest publication [9] for a detailed account. The resulting DT-Golog specification, can be used by the DT-Golog interpreter to identify *policies* in the form of conditional plans, that maximize expected utility in terms of the total expected satisfaction of the top level quality goal, considering the stochastic nature of actions. However, even without the DT-Golog interpreter, the result can be used for constructing simulations of the domain under consideration. This can be done by writing query routines and modules that simulate task execution, reconstruct the state of the system given a task history, inquire about task feasibility at a given state, and calculate the utility of a task at that state with respect to a quality of interest. In this way, the simulator exhibits a behavior that is compliant to the goal model. One of the uses of such goal-model driven development of simulators is reinforcement learning [13]. In recent work [11], we developed a component that allows such simulations through implementing a popular reinforcement learning (RL) interface, OpenAI's gym [14]. Using this component, off-the-shelf RL agents can directly use the specification as an alternative to DT-Golog reasoning.

The translator consists of three main layers. The input processing layer translates the XML into an input format-agnostic intermediate representation of the extended goal modeling language, which constitutes the middle layer. The spec generation layer reads the intermediate representation and translates it into the target specification. By separating the layers and introducing an intermediate specification, the translator can easily be adapted to alternative input formats, which can, in turn, be exports from different diagramming tools. Likewise, construction of translations to formalizations alternative to DT-Golog is independent of the tool used to develop the diagram.

3. Maturity and future work

The so far development constitutes an initial stage of a longer-term project for turning gReason into a comprehensive toolset for reasoning with action- and decision-theoretic goal models. In addition to ongoing quality assurance and documentation efforts as well as the strengthening of syntax validation facilities, both the language and the tool can be extended in various ways.

The language can be augmented in at least three ways. Firstly, modeling constructs can be added for informing RL training using the resulting specification. This includes continuous state variables and elements for describing episodic structure. Secondly, iStar actor elements can be re-introduced for modeling mult-actor problems using dependencies and delegations. Thirdly, options for describing state more expressively can be explored, by utilizing first order predicates, task and goal parameters, as well as domain objects.

At the same time the toolset can be extended to become more interoperable and appealing for different uses. Firstly, an input specification language can be defined to allow utilization of alternative diagramming tools for preparing the models. A possible starting point is iStarML [15] which will need to be extended with our additional constructs. Secondly, the intermediate model representation can be utilized for generating specifications that are alternative to DT-Golog. Assuming absence of non-determinism, a useful possibility is generation of HTN planning representations [7], enabling efficient minimum-cost identification of solutions to the goal model. Finally, the diagramming and automated analysis experiences can be combined in one user interface. This can come in the form of a plug-in to draw.io (or other open-source diagramming tool) that allows the calling of the automated reasoner from within the diagramming application and the immediate interpretation of its result into visual cues on the diagram.

4. Links

The shape library, translator code and executable, installation and usage directions, as well as video presentation can be found at https://github.com/cmg-york/gReason.

References

[1] E. S. K. Yu, Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering, in: Proc. of the 3rd IEEE International Symposium on Requirements Engineering (RE'97), Annapolis, MD, 1997, pp. 226–235.

[2] F. Dalpiaz, X. Franch, J. Horkoff, iStar 2.0 Language Guide, The Computing Research Repository (CoRR) abs/1605.0 (2016). URL: http://arxiv.org/abs/1605.07767. arXiv:1605.07767.

[3] J. Horkoff, Y. Yu, E. S. Yu, OpenOME: An Open-source Goal and Agent-Oriented Model Drawing and Analysis Tool, in: Proc. of the 5th International i* Workshop, 2011.

[4] D. Amyot, G. Mussbacher, S. Ghanavati, J. Kealey, GRL Modeling and Analysis with jUCMNav, in: Proc. of the 5th International i* Workshop, 2011.

[5] J. Pimentel, J. Castro, piStar Tool – A Pluggable Online Tool for Goal Modeling, in: Proc. of the 26th IEEE International Requirements Engineering Conference, 2018, pp. 498–499.

[6] Comparing the i* tools, http://istarwiki.org/, Retrieved: Sept. 14, 2023.

[7] S. Liaskos, S. McIlraith, S. Sohrabi, J. Mylopoulos, Representing and reasoning about preferences in requirements engineering, Requirements Engineering Journal (REJ) 16 (2011) 227–249.

[8] S. Liaskos, S. M. Khan, M. Soutchanski, J. Mylopoulos, Modeling and Reasoning with Decision-Theoretic Goals, in: Proceedings of the 32th International Conference on Conceptual Modeling, (ER'13), Hong-Kong, China, 2013, pp. 19–32.

[9] S. Liaskos, S. M. Khan, J. Mylopoulos, Modeling and reasoning about uncertainty in goal models: a decision-theoretic approach, Software & Systems Modeling 21 (2022) 1–24.

[10] M. Soutchanski, High-Level Robot Programming in Dynamic and Incompletely Known Environments, Ph.D. thesis, Department of Computer Science, University of Toronto, 2003.

[11] S. Liaskos, S. M. Khan, R. Golipour, J. Mylopoulos, Towards Goal-based Generation of Reinforcement Learning Domain Simulations, in: Proc. of the 15th International i* Workshop, 2022.

[12] draw.io (ver. 15.04.0), https://github.com/jgraph/drawio, Retrieved: 2022.

[13] R. S. Sutton, A. G. Barto, Reinforcement Learning: An Introduction, The MIT Press, 2018.

[14] Open AI Gym, 2022. URL: https://github.com/openai/gym.

[15] C. Cares, X. Franch, iStarML: Principles and Implications, in: Proc. of the 5th International i* Workshop, 2011.