

iTactic: A Goal Model Evaluation Tool to Support Strategic Decision-Making

Mohamed Abdel-Monem¹, Islam El-Maddah¹ and Hani Mahdi¹

¹Computer and Systems Engineering Department, Faculty of Engineering, Ain Shams University, Egypt

Abstract

Goal models proved their importance in several disciplines and fields of application. However, most goal modeling support tools are mainly biased toward requirements engineering. This paper presents a new conceptual tool “iTactic.” iTactic supports decision-making and tactics based on goal modeling. The tool defines a strategy in terms of a goal model using i* terminology. The employed evaluation method is based on the analogy to system models of control systems, accounting for input and output values at each point in time. iTactic goes beyond goal model evaluation and provides room for costs in its data model, providing a basis for comparing the expected success from different strategy implementation alternatives to the cost of these alternatives.

Keywords

Goal model, Goal model evaluation, Goal model evaluation tool, iStar, i*, Strategy evaluation, Decision-making tool, iTactic

1. Introduction

Policymakers establish both short and long-term strategies, usually sparking debates. Therefore, it is desirable to find metrics for the expected success of such strategies. Many available methodologies can help at this stage of decision-making, including Cost-Benefit Analysis, Payoff Tables, Decision Trees, and Analytic Hierarchy Process. Unfortunately, these methodologies cannot answer some questions regarding a specific strategy like: “What is the expected success?”, “When do we expect that success to happen?”, “What will be the cost?” and “Is this the optimal time and cost?” It is then needed to find a systematic way to predict the performance of a strategy. For example, it would be helpful to predict that Policy X will have a good start but a mediocre end; but Policy Y will appear cumbersome at the beginning; nevertheless, it will yield better results in the end.

In this paper, we introduce iTactic, a new tool to support strategic decision-making based on goal model evaluation. The framework and the theoretical bases are detailed in [1]. Other qualitative and quantitative approaches for goal model evaluations in both [2] and [3] assign initial labels or values representing the satisfaction level to leaf goals. These labels or values are then propagated upwards towards the final goal. Quantitative


iStar’22: The 15th International i Workshop, October 17th, 2022, Hyderabad, India*

✉ mmonem@gmail.com (M. Abdel-Monem); islam.elmadah@eng.asu.edu.eg (I. El-Maddah);

hani.mahdi@eng.asu.edu.eg (H. Mahdi)

ORCID 0000-0002-4532-4477 (M. Abdel-Monem)

© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

approaches based on fuzzy logic have been presented in [4] and [5]. A quantitative approach for evaluating KAOS-based goal models was introduced based on a probabilistic model to represent partial satisfaction of the goals presented in [6]. GeNIe Modeler[7] provides a similar concept to goal model evaluation but also relies on a probabilistic model and does not support the goals' nodes to form cycles.

These mentioned satisfaction analysis techniques consider only what value the final goal will eventually take, ignoring the time profile of satisfaction levels of both leaf goals (the inputs) and the root goal (the output). Additionally, these techniques ignore the nature of specific goal satisfaction dynamics, e.g., the required time for goals to be satisfied after their precedents were fully satisfied. Other research works, e.g. in [8] and [9], consider the time domain for goal satisfaction levels and even for the structure of the goal model itself. Nevertheless, these researches still don't account for the natural propagation delay in goal satisfaction.

In the selected framework for strategy evaluation, a strategy is modeled in terms of an i^* -like goal model. The strategy's final goal and other intermediate goals are treated as i^* softgoals. Tasks are the leaf nodes in the goal model, and they are the original source of the contribution. This contribution is specified for each task in terms of a pre-estimated curve named the success curve. The success curve defines how the task will develop success. Another curve to be specified for each task is the cost curve which also specifies how the cost will be distributed along the task execution time.

In iTactic, positive and negative contribution relationships are the only supported relationships among goals or from tasks to goals. The framework accepts the contribution relationships to form cycles in the goal model. The individual success of a task propagates to participate in the success curve of the final goal. On the other hand, the cost curve of the final goal can be computed algebraically. Comparing the cost and success curves for the final goal of different policies provides a means for choosing the best decision to take.

2. Tool Objectives

iTactic is used to prove the novel concept in [1]. That concept ties the control system model to the goal modeling. iTactic is designed specifically to address:

1. Graphical editing goal models in accordance with i^* terminology.
2. Extending the i^* softgoal data model to accommodate the extended attributes: softgoal responsiveness and weights from the contributing nodes.
3. Extending the i^* task data model to accommodate the entry of cost and success curves.
4. Graphical data entry of the extended attributes.
5. Providing an interactive graphical user interface to start the evaluation of different combinations of tasks based on task inclusion or exclusion.
6. Evaluating the resultant success curve of the final goal based on state-space representation according to the methodology that is detailed in [10].
7. Calculating the resultant cost curve of the final goal.
8. Visualizing the resultant cost/success curves for the goal model final softgoal.

3. Internal Design

Based on the analogy of goal models to system models of control systems, the used goal model evaluation is based on the state-space representation. This allows the accounting of values at each point in time for the inputs and the output. The details of the method along with the used algorithms are stated in [10].

State-space representation has a mature open-source implementation in the signal module of SciPy which is a Python library for scientific computing. This drove iTactic to use Python as the foundation programming language. Python also provides a myriad of facilities for matrix representation, graphical user interface, and data visualization. PyQt [11] is the python binding of the prominent graphical user interface toolkit Qt which has been adopted by iTactic and used extensively. All package dependencies of iTactic are listed in Table 1.

Table 1
iTactic package dependencies

	Dependency	Version	Purpose
1	PyQt	5.15.7	Graphical user interface based on the Qt library
2	NumPy	1.23.0	Computation library, used mainly for matrices implementation and manipulation
3	SciPy	1.8.1	Python library for scientific and technical computing, used for state-space simulation
4	matplotlib	3.5.2	Python library for Graph visualization for displaying various curves

iTactic attempts to separate the presentation layer of the application from the internal data model presentation and computations layer. This is achieved by the segregation of classes into `gui` namespace and `model` namespace. Among the classes in the `gui` namespace is the `gui.Scene` class, which inherits `PyQt5.QtWidgets.QGraphicsScene`. The primary task of this class is to populate the graphical nodes and relations (the `gui.graphics.*` classes inheriting from `QGraphicsItem`) from the `model.Document` class. The class diagram of the graphical elements of the goal model is shown in Figure 1.

From the class diagram it can be indicated that while the tool supports actor nodes, it does not support dependencies in its current version. Therefore, it cannot currently be used for Strategic Dependency models as indicated in [12].

JSON format has been chosen for saving the model as computer files. It allows for external editing by text editors and gives chance for possible interoperability with other modeling tools.

For handling mouse events inside the `QGraphicView` which is the widget that hosts the goal model, an object-oriented approach is used based on a concept of an interactor. An interactor is an object that handles user events like mouse and keyboards. The naming matches a similar concept used by ILOG Views, which was a famous object-oriented C++ toolkit for graphical user interface and data visualization in the 1990s and 2000s. Another notable task of the `gui.Scene` is dispatching keyboard and mouse events to the `gui.interactors.Interactor` subclasses that define the user interface behavior. The

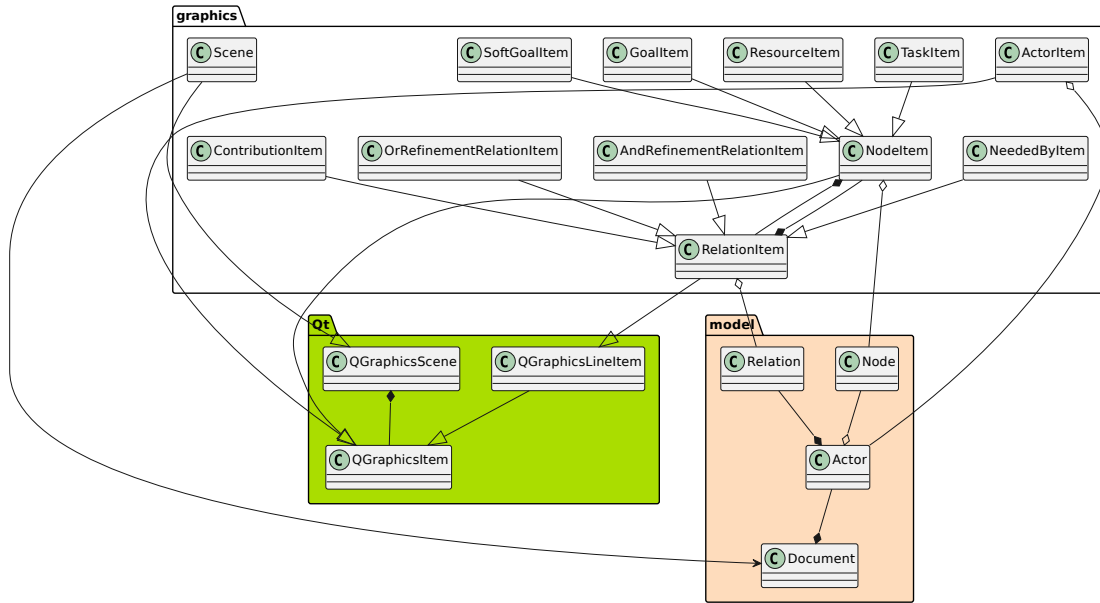


Figure 1: The Class Diagram of iTactic Graphical Elements

class diagram of the implemented interactors for manipulating various goal model elements is shown in Figure 2.

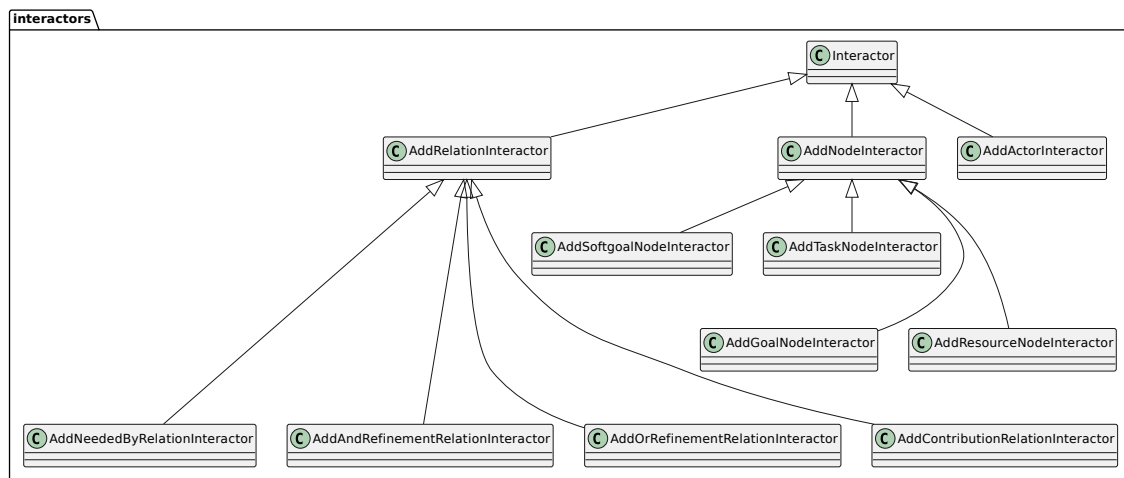


Figure 2: The Class Diagram of Interactors

4. Using iTactic

The general steps for using iTactic are:

1. Develop the goal model that represents the strategy to achieve the final goal.
2. For each goal, define the responsiveness parameter and the weights of contributions from other goals or tasks. Weights should sum up to the value of 1.0.
3. For each task, determine the best cost and success curve parameters that best match its expected performance. The supported types are Linear and Sigmoid. Additional parameters are required according to the type as seen in Figure 3:
 - Linear type: The “Start” parameter specifies when the linear period of cost/success starts and the “End” parameter specifies when that period ends.
 - Sigmoid type: The “Hump” value of the curve, which specifies the center of the curve that corresponds to the point in time with the maximum rate of change in the curve, and to set the “Speed” value to specify how sharp the curve will look like
4. Start evaluation of the goal model and define strategy implementation alternatives by selecting combinations of tasks as seen in Figure 4). The simulation is instantly re-executed, and the output success and cost curves are shown allowing the user to review the expected cost and success curves of the final goal for these combinations.

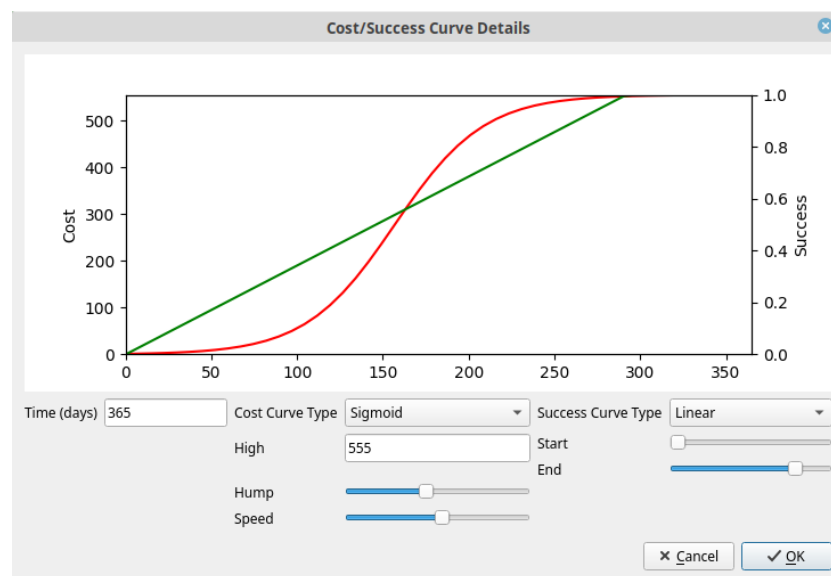


Figure 3: The dialog box of the cost/success details of a softgoal

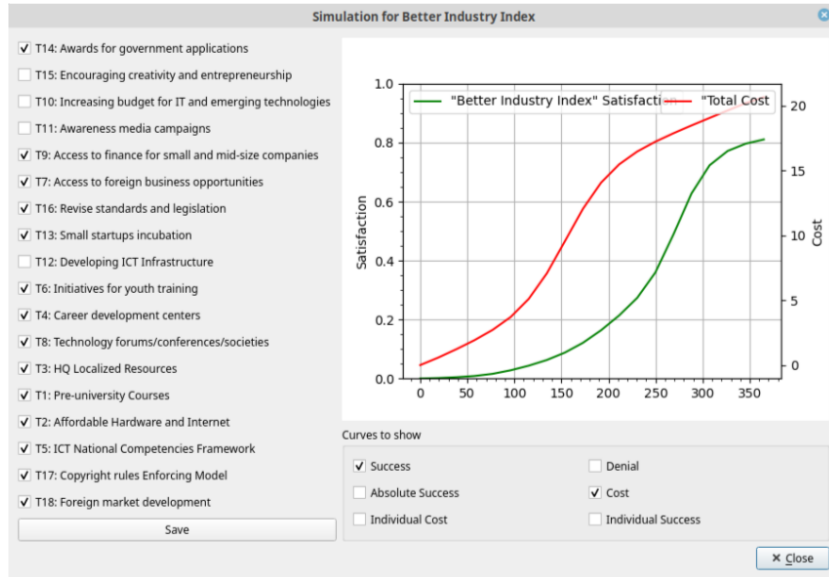


Figure 4: The dialog box of the simulation dialog

5. Conclusions and Ongoing Work

iTactic is a tool that is capable of editing i^* goal models. It also introduced novel evaluation capabilities that facilitates identifying strategy implementation alternative by task inclusion/exclusion. The output of an alternative in iTactic is represented as the cost/success curves of the final goal of the goal model. The resultant cost/success curves carry insights about the expected performance of the strategy implementation alternatives, hence supporting decision-making.

However, the current version supports only the basic parts of the i^* language. Currently, it supports all node types but the relationship types are limited to contributions. This is due to the fact that the employed evaluation algorithm uses the state-space representation which requires linear relations only. A possible future work can address this limitation possibly by linearization techniques.

The underlying method for goal model evaluation supports any time series for the cost/success curves of tasks. However, the current version of the tool supports graphical-based data entry of only curves that are either linear or sigmoid-like. More options for cost/success curves since are currently being studied to be supported in a next version. Other planned enhancements of the tool include modeling and handling of interdependencies among tasks, and implementing cost/success curves' alternatives for tasks. Ongoing work is currently being done also to provide internal interoperability with other i^* tools by importing and exporting files in common formats, e.g. iStarML [13] or piStar's JSON [14].

References

- [1] M. Abdel-Monem, M. Mansour, I. El-Maddah, H. Mahdi, Strategy assessment using goal models: Software industry as a case study example, in: Proceedings of the 14th International iStar Workshop (iStar 2021), CEUR Workshop Proceedings, 2021, pp. 1–7.
- [2] L. Chung, B. A. Nixon, E. Yu, J. Mylopoulos, Non-Functional Requirements in Software Engineering, volume 5 of *International Series in Software Engineering*, Springer US, 2000. doi:10.1007/978-1-4615-5269-7.
- [3] P. Giorgini, J. Mylopoulos, E. Nicchiarelli, R. Sebastiani, Reasoning with goal models, in: S. Spaccapietra, S. T. March, Y. Kambayashi (Eds.), *Conceptual Modeling — ER 2002*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2003, pp. 167–181.
- [4] C. M. Subramanian, A. Krishna, A. Kaur, R. P. Gopalan, Quantitative reasoning of goal satisfaction in the i* framework, in: Proceedings of the 27th International Conference on Software Engineering and Knowledge Engineering, KSI Research Inc. and Knowledge Systems Institute Graduate School, 2015, pp. 666–669. doi:10.18293/SEKE2015-158.
- [5] G. Chatzikonstantinou, K. Kontogiannis, Efficient parallel reasoning on fuzzy goal models for run time requirements verification, *Software & Systems Modeling* 17 (2018) 1339–1364. URL: <https://doi.org/10.1007/s10270-016-0562-9>. doi:10.1007/s10270-016-0562-9.
- [6] E. Letier, A. van Lamsweerde, Reasoning about partial goal satisfaction for requirements and design engineering, *SIGSOFT Softw. Eng. Notes* 29 (2004) 53–62. URL: <https://doi.org/10.1145/1041685.1029905>. doi:10.1145/1041685.1029905.
- [7] L. BayesFusion, Genie modeler, User Manual. Available online: <https://support.bayesfusion.com/docs/> (accessed on 21 October 2019) (2017).
- [8] G. Mussbacher, et al., Timedgrl: specifying goal models over time, in: 2016 IEEE 24th International Requirements Engineering Conference Workshops (REW), IEEE, 2016, pp. 125–134.
- [9] A. M. Grubb, M. Chechik, Formal reasoning for analyzing goal models that evolve over time, *Requirements Engineering* 26 (2021) 423–457.
- [10] M. Abdel-Monem, I. El-Maddah, H. K. Mahdi, Goal model evaluation based on state-space representation, *IEEE Access* 8 (2020) 205111–205123. doi:10.1109/ACCESS.2020.3037210.
- [11] Riverbank Computing Limited, PyQt, 2022. URL: <https://www.riverbankcomputing.com/software/pyqt/>.
- [12] E. Yu, Modelling Strategic Relationships for Process Reengineering, Ph.D. thesis, Graduate Department of Computer Science, University of Toronto, 1995.
- [13] C. Cares, X. Franch, A. Perini, A. Susi, Towards interoperability of i* models using iStarML, *Computer Standards & Interfaces* 33 (2011) 69–79.
- [14] J. Pimentel, J. Castro, piStar tool—a pluggable online tool for goal modeling, in: 2018 IEEE 26th International Requirements Engineering Conference (RE), IEEE, 2018, pp. 498–499.