A Qualitative, Interactive Evaluation Procedure for Goal- and Agent-Oriented Models

Jennifer Horkoff[1], Eric Yu[2]

[1]Department of Computer Science, [2]Faculty of Information,
University of Toronto, Canada
jenhork @ cs.utoronto.ca, yu @ ischool.utoronto.ca

Abstract. Applying systematic analysis procedures to early requirements models can test the satisfaction of stakeholder goals and facilitate an evaluation of design alternatives. We introduce a qualitative and interactive model evaluation procedure for goal and agent-oriented models. In applying the procedure to a variety of case studies we found that the interactive nature of the procedure prompts model iteration, producing higher quality models.

1 Introduction

Goal- and agent-oriented modeling frameworks, such as i* [1], have been used to perform “Early” Requirements Engineering (ERE), advocating for modeling and exploration of the socio-technical domain before focusing on detailed system functionality. In order to analyze such models, systematic analysis procedures are needed, considering the chain of effects, propagating among goals and functionalities throughout the network in a consistent way. The primary aim of such analysis is to determine whether stakeholder’s goals can be achieved, given domain assumptions.

The informal and incomplete nature of goal models calls for analysis procedures which are interactive, qualitative, and simple to apply. We introduce such a procedure for goal- and agent-oriented models. We describe the specifics of the procedure in terms of the i* Framework, although the procedure can potentially be applied to other similar models. The procedure is adapted from an evaluation procedure originating from the NFR Framework [2]. We expand on this work by clearly describing the application of the procedure, taking into account agent-related features, and examining the procedure’s role in model iteration. We discuss the procedure’s benefits by briefly describing its application to various case studies.

2 Related Work

As the flexibility of goal- and agent-oriented modeling allows application in many stages of system development, different analysis approaches may be more appropriate for different stages. For early-stage modeling, where specific quantitative measures are scarce, qualitative, interactive evaluation is appropriate. Evaluation in the NFR

Framework propagates qualitative labels throughout a Softgoal Interdependency Graph (SIG), prompting the user to resolve conflicts [2]. Case study experience has found human intervention in the NFR procedure to be too restrictive, automatically propagating conflicts and unknown values when the evaluator would prefer to have input. Previous work assumed that the NFR procedure could be easily extended for use with i*, without describing extensions to support additional syntax (e.g., [3]).

Giorgini et al. have introduced qualitative and quantitative procedures for goal model analysis which separately propagate negative and positive evidence, are fully automated, and work in a forwards and backwards direction [4]. Recent work on GRL [5], a variant of i*, includes several evaluation methods, ranging from quantitative to qualitative. The full automation in these procedures does not give the evaluator freedom to make decisions in the presence of conflicting, partial or unknown information. The hard-coded rules used to resolve softgoals often result in the proliferation of conflicts or partial values. Where quantitative values are not derived from domain measures, they can be viewed as fine-grained qualitative judgments. There is danger that users may place an undeserved amount of confidence in the computed results, associating them with mathematical precision.

For later stages of system analysis, where quantitative information is known and where models are relatively stable, fully automated and quantitative evaluation can be appropriate. Example methods more appropriate for this later-stage evaluation include evaluation in the KAOS Framework [6], the analysis of property metrics over the structure of goal- and agent- oriented models [7], planning and simulation over goal- and agent-oriented models [8], and checks of properties over goal models [9].

3 The Qualitative, Interactive Evaluation Procedure

The procedure is designed to be applied either manually or semi-automatically. Here, we focus on describing the procedure so that it can be applied manually. In order to concretely describe the procedure, we apply it to the i* Framework [1]. This Framework facilitates exploration of the system domain, emphasizing social aspects by providing a graphical depiction of system actors, their intentions, dependencies, and alternatives. The social aspect is represented by *actors*, who depend upon each other for the accomplishment of *tasks*, the provision of *resources*, and the satisfaction of *goals* and *softgoals*, goals without clear-cut criteria for satisfaction. Actors can be "opened-up" using *actor boundaries* containing the desired elements (goals, softgoals, tasks, and resources) of the actor. The interrelationships between elements inside an actor are depicted with *Decomposition* links, showing elements necessary to accomplish a task; *Means-Ends* links, showing alternative tasks to accomplish a goal; and *Contribution* links, showing the effects of elements on softgoals. Positive/negative contributions representing evidence which is strong enough to satisfy/deny a softgoal are represented by *Make*/*Break* links. Contributions that are not sufficient to satisfy/deny a softgoal are represented by *Help*/*Hurt* links.

Procedure Overview: The proposed procedure starts with an analysis question such as "How effective is this design option with respect to the desired goals?" The procedure makes use of a set of qualitative evaluation labels, assigned to elements to express their degree of satisfaction or denial. The process starts by assigning initial label values to model elements representing the analysis question. These values are propagated through the model links using defined rules. Human judgment is needed when multiple conflicting or partial values must be combined to determine the satisfaction or denial of a softgoal. The final satisfaction and denial values for the elements of each actor are analyzed in light of the original question. An assessment is made as to whether the design choice is satisficed ("good enough"), likely stimulating further analysis and potential model refinement. More detail can be found in [10].

Detailed Steps: We first provide the steps of the evaluation procedure, followed by detailed explanation of the concepts.

1. **Initiation:** The evaluator decides on an analysis question and applies the initial evaluation labels to the model. The initial values are added to a label queue.

Iteratively, until the label queue is empty or a cycle is found:

2. **Propagation:** The evaluation labels in the label queue are propagated through all outgoing adjacent model links. Resulting labels propagated through non-contribution links are placed in the label queue. Results propagated through contribution links are placed into a "label bag" for that element.
3. **Softgoal Resolution:** Label bags are manually resolved, producing a single result label which is added to the label queue.

4. **Analysis:** The final results are examined to find an answer to the analysis questions. Issues with the model can be discovered, prompting further analysis.

Model Syntax: The procedure assumes that models are well-formed. However, as propagation is dependent on link and not element type, most models can be evaluated.

Qualitative Evaluation Labels: We adopt the qualitative labels used in NFR evaluation (Table 1). The *(Partially) Satisfied* label represents the presence of evidence which is *(insufficient)* sufficient to satisfy an element. *Partially denied* and *denied* have the same definition with respect to negative evidence. *Conflict* indicates the presence of both positive and negative evidence of roughly the same strength. *Unknown* represents the situation where there is evidence, but its effect is unknown. We introduce the "None" label to indicate a lack of any label. We adopt the use of partial labels for non-softgoals to allow for greater expressiveness.

Initial Evaluation Values: In order to start an evaluation of a model, a set of initial evaluation values must be placed on the model, reflecting a particular analysis question, comprising Step 1 of the procedure. Often, initial values are placed on "leaf" elements in the model, elements that do not receive input from other elements. However, initial values can also be placed on non-leaf elements. In this case, we avoid overriding the initial labels with subsequent propagation.

Evaluation Propagation Rules: We define rules in order to facilitate a standard propagation of values given a link type and contributing label in Step 2 of the procedure. Here, we must define how evaluation values should be propagated through link types that are in i* but not in the NFR framework, namely, *Means-Ends*, *Decomposition*, and *Dependency* links. The nature of a **Dependency** indicates that if the *dependee* is satisfied then the *dependum* will be satisfied, and so will the *depender*. **Decomposition** links depict the elements necessary to accomplish a task, indicating the use of an AND relationship, selecting the "minimum" value amongst all of the values. Similarly, the **Means-Ends** link depicts the alternative tasks which are able to satisfy a goal, indicating an OR relationship, taking the maximum values of elements in the relation. To increase flexibility, the OR is interpreted to be inclusive. We expand the order of the values presented in the NFR Framework to allow for partial values, producing: None < ✗ < ✗• < ? < ⚡ < ✓• < ✓

We adopt the **contribution** links propagation rules from the NFR procedure. These rules intuitively reflect the semantics of contribution links. Note that the “None” label is not propagated or placed in the label queue.

Table 1. Propagation Rules Showing Resulting Labels for *Contribution* Links

Source Label		***Contribution* Link Type**						
	Name	**Make**	**Help**	**Some+**	**Break**	**Hurt**	**Some-**	**Unkn.**
✓	Satisfied	✓	✓•	✓•	✗	✗•	✗•	?
✓•	Partially Satisfied	✓•	✓•	✓•	✗•	✗•	✗•	?
⚡	Conflict	⚡	⚡	⚡	⚡	⚡	⚡	?
?	Unknown	?	?	?	?	?	?	?
✗•	Partially Denied	✗•	✗•	✗•	✓•	✓•	✓•	?
✗	Denied	✗	✗•	✗•	✓•	✓•	✓•	?

Resolving Multiple Contributions: Softgoals are often the recipient of multiple contribution links. We adopt the notion of a “Label Bag” from the NFR Framework, used to store all incoming labels for a particular softgoal. Labels in the bag are combined into a single label in Step 3, either by identifying specific cases where the label can be determined without judgment (Table 2), or by human judgment.

Table 2. Cases where Overall Softgoal Labels can be Automatically Determined

Label Bag Contents	**Resulting Label**
1. The bag has only one label : {<v, e_s>}	the label: v
2. The bag has multiple full labels of the same polarity, and no other labels. Ex: {✓, ✓, ✓} or {✗, ✗}	the full label: ✓ or ✗
3. All labels in the bag are of the same polarity, and a full label is present. Ex: {✓•, ✓, ✓•} or {✗, ✗•}	the full label: ✓ or ✗
4. The previous human judgment produced ✓ or ✗, and a new contribution is of the same polarity	the full label: ✓ or ✗

Human Judgment in Evaluation: Human judgment is used to decide on a label for softgoals in Step 3, for the cases not covered in Table 2. Human judgment may be as simple as promoting partial values to a full value, or may involve combining many sources of conflicting evidence. When making judgments, domain knowledge related to the destination and source elements should be used.

Combinations of Links: Elements in i* are often the destination of multiple types of links. “Hard” links (Decomposition, Means-Ends and Dependency) are combined using an AND of the final results of each link type. If Contribution and Dependency links share the same destination, the result of the Dependency links are treated as a Make contribution, considered with the other contributions in the label bag.

Incomplete Labels: In the procedure, information present in each step is propagated, even if this information in incomplete, i.e., other incoming contributions are missing. As a result, the same element may receive multiple evaluation labels in one evaluation, and the same softgoal may require human judgment multiple times.

Detecting Cycles: Goal models often contain cycles, values which indirectly contribute to themselves and may cause fluctuating values. Experience has shown that the presence of cycles becomes apparent to the evaluator after a few iterations.

Example: We provide a simplified example from the Trusted Computing (TC) Case Study [10] in Fig. 1. This model depicts a simplistic view of the TC domain, showing the intentions of the PC User, the PC Product Provider and the Data Pirate. In our example evaluation, we ask: "If the PC User Obtains PC Products from the Data Pirate, how does this affect the PC Product Provider's ability to Sell PC Products for Profit?" Initial values are circled and human judgment is annotated in the model.

In this example, when PC Products are Obtained from the Data Pirate, PC Products are Obtained Affordably, but the PC Product Provider does not Sell PC Products for Profit. Further rounds of evaluation and model iteration are needed. In this simple model, analysis results may be apparent without applying explicit procedures. However, in larger goal models results are not apparent and are difficult to derive consistently.

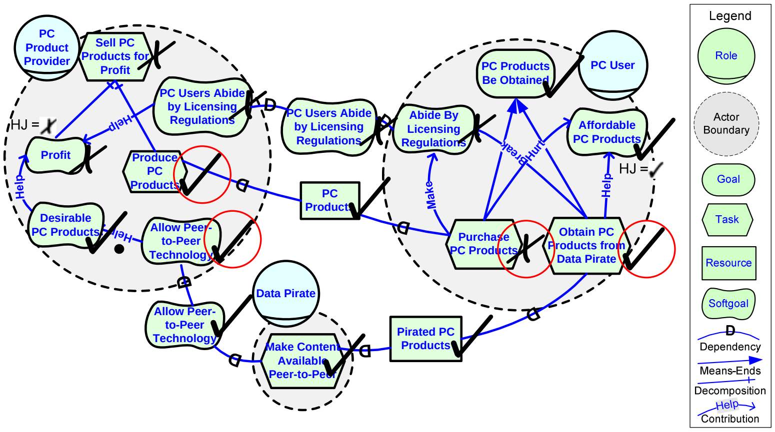

Fig. 1. Simplified TC Example showing Final Evaluation Results

4 Discussion, Conclusions and Future Work

Benefits of applying the procedure described in this work include both the ability to answer strategic questions and the means to iterate upon and improve the quality of the model and subsequent domain understanding. For example, in a case study involving a large social service organization [12], the procedure was applied manually to large models to evaluate the effectiveness of various technologies. Results showed that a wiki was not effective in satisfying the goals of the organization, while a discussion forum showed more promise. In another example, when applying the procedure to the TC case study, the model appeared to be sufficiently complete; however, analysis of the TC Opponent point of view revealed that PC Users would not buy TC Technology. However, the makers of TC Technology must have envisioned some way in which users would accept their product. This result prompted model changes, adding factors such as product lock-in.

In this work, we have introduced a relatively simple analysis procedure which builds on the NFR procedure, providing specific instructions for manual application, and expanding the algorithm to deal with i*-specific constructs. We have highlighted the benefits of such a procedure, including analysis, and model iteration.

This work can be expanded in several ways. For example, evaluating the satisfaction of actors, as in [5], and expanding to "top-down" or backwards analysis [13]. A version of the procedure is currently being re-implemented in the OpenOME Tool [14]. We are in the process of administering experiments to further test the procedure's ability to facilitate analysis and provoke model iteration.

Acknowledgements: Financial support has been provided by Bell University Laboratories and the Ontario Graduate Scholarship Program.

References

1. Yu, E.: Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering. In 3rd IEEE International Symposium on Requirements Engineering (RE'97), pp. 226--235. IEEE Press, New York (1997)
2. Chung, L., Nixon, B.A., Yu, E., Mylopoulos, J.: Non-Functional Requirements in Software Engineering. Kluwer Academic Publishers, Norwell, MA (2000)
3. Liu, L., Yu, E.: Designing Information Systems in Social Context: A Goal and Scenario Modelling Approach. Information Systems. 29(2), 187--203 (2004)
4. Giorgini, P., Mylopoulos, J., Sebastiani, R.: Simple and Minimum-Cost Satisfiability for Goal Models. In: Persson, A., Stirna, J. (eds) CAiSE 2004. LNCS, vol. 3084, pp. 20-3-5. Springer, Heidelberg (2004)
5. Amyot, D., Ghanavati, S., Horkoff, J. Mussbacher, G., Peyton, L., Yu, E.: Evaluating Goal Models within the Goal-oriented Requirement Language. (Submitted manuscript).
6. Letier, E., Lamsweerde, A. van: Reasoning about Partial Goal Satisfaction for Requirements and Design Engineering, In: 12th ACM International Symposium on the Foundations of Software Engineering (FSE'04), pp. 53--62. ACM, New York (2004)
7. Franch, X.: On the Quantitative Analysis of Agent-Oriented Models: In Dubois, E., Pohl, K. (eds) CAiSE'06. LNCS, vol. 4001, pp. 495--509. Springer, Heidelberg (2006)
8. Gans, G., Jarke, M., Lakemeyer, G., Schmitz, D.: Deliberation in a Modeling and Simulation Environment for Inter-organizational Networks. In: Eder J., Missikoff M. (eds), Proceedings CAiSE'03. LNCS, vol. 2681, pp. 242--257, Springer, Heidelberg (2003)
9. Fuxman, A., Pistore, A., Mylopoulos, J., Traverso, P.: Model Checking Early Requirements Specifications in Tropos. In Fifth IEEE International Symposium on Requirements Engineering (RE01). pp. 174-181. IEE Press, New York (2001)
10. Horkoff, J.: An Evaluation Algorithm for the i* Framework. Master's Thesis, Department of Computer Science, University of Toronto (2006)
11. Horkoff, J., Yu, E., Liu, L.: Analyzing Trust in Technology Strategies. In: International Conference on Privacy, Security and Trust (PST 2006). pp. 21--32. (2006)
12. Strohmaier, M., Horkoff, J., Yu, E., Aranda, J., Easterbrook, S.: Can Patterns improve i* Modeling? Two Exploratory Studies. In: Paech, B., Rolland, C. (eds) REFSQ'08. LNCS, vol. 5025, pp. 153--167. Springer, Heidelberg (2008)
13. Horkoff, J., Yu, E.: Qualitative, Interactive, Backward Analysis of i* Models. In: 3rd International i* Workshop. pp. 43—46, CEUR-WS.org (2008)
14. OpenOME, https://se.cs.toronto.edu/trac/ome/wiki