

# Qualitative, Interactive, Backwards Analysis of i\* Models

Jennifer Horkoff<sup>1</sup>, Eric Yu<sup>2</sup>

<sup>1</sup>Department of Computer Science, University of Toronto

<sup>2</sup>Faculty of Information Studies, University of Toronto

{[fjenhork@cs.utoronto.ca](mailto:fjenhork@cs.utoronto.ca), [yu@fis.utoronto.ca](mailto:yu@fis.utoronto.ca)}

**Abstract.** Deriving the full benefits of i\* models requires analysis and iteration beyond initial construction. This work outlines a procedure which allows backwards (top-down), qualitative, interactive analysis of i\* models using SAT solving techniques. This approach expands on work in goal model reasoning, expressing i\* models and qualitative evaluation values as a SAT problem. The nature of the interaction with the user will be explored. Future work will include an implementation of this procedure into the Eclipse-based OpenOME tool and application to a detailed case study.

## 1 Introduction

Much attention in the requirements modeling community has been paid to the use of i\* models as tools for early requirements analysis. Deriving the full benefits of such models requires analysis and iteration beyond initial construction. To this end, various i\* analysis methodologies and techniques have been proposed.

Previously proposed techniques for i\*, or related goal model analysis, provide approaches which range in various dimensions including the level of automation, the direction of analysis, and the role of human intervention. Techniques introduced for reasoning with goal models allow qualitative evidence to be propagated in a backwards (top-down) manner, in a fully automated way using SAT solving techniques, and by separating positive and negative evidence [1]. In contrast, a method introduced for analysis of i\* models allows qualitative evidence to be propagated in a forwards (bottom-up) manner, requiring interaction with the user to resolve combinations of positive and negative evidence ([2], [3]).

This ongoing work attempts to combine these two approaches, employing SAT techniques to allow backwards, qualitative, interactive analysis of i\* models. This will allow for a new type of analysis to be used with i\* models, providing the ability to ask new types of analysis questions.

## **2 Objectives of the Research**

In order to develop a backwards, qualitative, interactive analysis procedure for the i\* Framework, we intend to do the following:

- Expand work which expresses goal model analysis as a SAT problem [1] in order to express i\* model analysis as a SAT problem.
- Determine what information is needed from the user, including how this information can be used, stored and reused.
- Analyze computational issues such as correctness and termination.

This work will briefly describe our progress toward these specific objectives.

## **3 Scientific Contributions**

In the following section, we outline a procedure and framework which combines the work of [1] and [2], allowing for backwards, qualitative, interactive, i\* analysis. We define the use of a SAT solver in an iterative procedure then briefly sketch how we express the required formal constructs. Finally, we provide an example for illustrative purposes.

### **3.1 Iterative Procedure**

Generally, the procedure iteratively runs a SAT solver, given a SAT formula which encodes the forward and backward propagation of i\* constructions, including target evaluation values for certain elements, and input elements, whose values, through propagation, produce these target values (line 1). The representation of i\* models in a SAT formula allows elements to have both positive and negative evidence, as in [1]. However, to follow the conventions of [2], such evidence must be combined using human judgment. Therefore, once the SAT procedure is run with the formula encoding an i\* model (line 3), the results are checked for softgoals that would normally require human judgment (have some combination of positive, negative, partial, conflicting or unknown evidence) (line 9). Starting with the softgoals which are closest to the targets, the procedure asks the user for one or more combinations of contributing elements which would produce the values for these softgoals required by the SAT solver (lines 10-12). These decisions are encoded in new clauses which are added to the SAT formula, replacing the previous clauses which applied to the softgoal (line 13). The entire procedure is run again (lines 14, 3). If the newly entered decisions do not result in a satisfying result, the user is prompted again to enter further possible combinations, if such combinations exist (line 5). Finally, if a satisfying assignment is found with SAT results such that no softgoals require human judgment, the procedure completes successfully, providing the necessary input values needed to produce the desired analysis values for the target elements (lines 15-16). The procedure is outlined in the high-level pseudocode shown in Fig. 1.

1	Create $\Phi$ , SAT Formula
2	Start
3	Run SAT( $\Phi$ )
4	If no satisfying assignment
5	If the user has been previously prompted for judgment, starting from the most recent, for each such softgoal prompt again, if new input is given
6	Update SAT Formula with new user input
	Goto Start
7	Else
	No satisfying assignment found, END
8	Else, a satisfying assignment is found
9	If human judgment is required for any softgoal
10	Find the softgoals requiring human judgment closest to target elements
11	Find the necessary analysis values for these softgoals according to SAT
12	Prompt the user for combinations of contributing analysis values which would achieve these necessary values
13	Update SAT Formula with these values
14	Goto Start
15	Else, human judgment not needed on any softgoal
16	Success, return SAT results

Fig. 1. High-Level Summary of Analysis Procedure

### 3.2 Expressing i\* Propagation as a SAT Formula

In order for the procedure outlined in the above section to work correctly, we must be able to represent the propagation of an i\* model in a SAT formula. Similar to the approach in [1], we must formally define the construction of an i\* model, define axioms for forward and backward propagation and describe how this information is combined into a SAT formula. The details of these formalizations are omitted due to space constraints. Generally, the formalizations provided in [1] are modified to take into account i\* syntax which is not used in goal graph constructs, including dependencies and differing types of contribution links. Currently the presence of actors or actor boundaries does not effect propagation as described in [2]. In addition, the formalizations are adjusted to account for the additional evaluation values of conflict and unknown. Once the axioms for forward and backward propagation are adjusted for use with i\*, the SAT formula can be constructed in the same manner as in [1]. However, if more complex i\* syntax, such as a mixture of incoming link types, were to be considered this formula may need to be altered.

### 3.2 Example

Consider the simple model in Fig. 2, representing password implementation choices. Take the example of the user choosing the partial satisfaction of Attract Users as a Target, with input goals of the tasks related to password implementation. During the first iteration of the procedure the user would be told that Attract Users

needs to be partially satisfied and asked to input one or more combinations of analysis values for Security and Usability which produces this value. If the user indicates that both these elements need to be satisfied, the procedure will come back to the user asking, individually, what combinations of the input tasks would cause Security and then Usability to be satisfied. Assuming the user does not manually catch the conflict, she might say that Restrict Structure of Password and Ask for Secret Question need to both be Satisfied for the first question, and then Denied and Satisfied, respectively, for the second question. The procedure will then indicate that a satisfying assignment was not found, and return to the previous questions asking for further viable combinations for Security and Usability, and then, if no viable combinations are found for these elements, for Attract Users. After several iterations, the user may come to an agreeable combination, for example, with both tasks satisfied Security is fully satisfied while Usability has conflicting value, or, a satisfying assignment may not be found, indicating that the desired target values are not feasible.

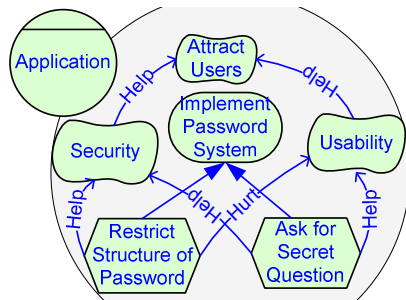


Fig. 1: Example i\* Model and Forward Evaluation

## 4 Conclusions, Ongoing and Future Work

Preliminary investigations indicate that it is possible to implement a reasonable qualitative, backward, interactive analysis procedure for i\*; however much future work is needed. Currently, we are developing proofs of correctness for the algorithm and SAT formula, including termination. We are also investigating ways to reuse the information provided by the users in future rounds of analysis, as well as ways to make the user interaction more user-friendly. Future work will include an implementation of this procedure into the Eclipse-based OpenOME tool and application to a detailed case study, testing its practical applicability.

## References

1. Sebastiani, R., Giorgini, P., Mylopoulos, J.: *Simple and Minimum Cost Satisfiability for Goal Models*. CAiSE'04 (2004)
2. Horkoff, J.: *Using i\* Models for Evaluation*. MSc Thesis, University of Toronto (2006)
3. Horkoff, J., Yu, E.: *Using the i\* Evaluation Procedure for Model Analysis and Quality Improvement*. Presented at the Second i\*/Tropos Workshop (2005)