Iterative, Interactive Analysis of Agent-Goal Models for Early Requirements Engineering

by

Jennifer Marie Horkoff

A thesis submitted in conformity with the requirements
for the degree of Doctor of Philosophy

Department of Computer Science
University of Toronto

Iterative, Interactive Analysis of Agent-Goal Models for Early Requirements Engineering

Jennifer Horkoff

Doctor of Philosophy

Department of Computer Science
University of Toronto

2012

Abstract

Conceptual modeling allows abstraction, communication and consensus building in system development. It is challenging to expand and improve the accuracy of models in an iterative process, producing models able to facilitate analysis. Modeling and analysis can be especially challenging in early Requirements Engineering (RE), where high-level system requirements are discovered. In this stage, hard-to-measure non-functional requirements are critical; understanding the interactions between systems and stakeholders is a key to system success. Goal models have been introduced as a means to ensure stakeholder needs are met in early RE. Because of the high-level, social nature of early RE models, it is important to provide procedures which prompt stakeholder involvement (interaction) and model improvement (iteration). Most current approaches to goal model analysis require quantitative or formal information that is hard to gather in early RE, or produce analysis results automatically over models. Approaches are needed which balance automated analysis over complex models with the need for interaction and iteration.

This work develops a framework for iterative, interactive analysis for early RE using agent-goal models. We survey existing approaches for goal model analysis, providing guidelines using domain characteristics to advise on procedure selection. We define requirements for an agent-goal model framework specific to early RE analysis, using these requirements to evaluate the appropriateness of existing work and to motivate and evaluate the components of our analysis framework. We provide a detailed review of forward satisfaction procedures, exploring how different model interpretations affect analysis results. A survey of agent-goal variations in practice is used to create a formal definition of the i* modeling framework which supports sensible syntax variations. This definition is used to precisely define analysis procedures and concepts throughout the work. The framework consists of analysis procedures, implemented in the OpenOME requirements modeling tool, which allow users to ask "What if?" and "Is this goal achievable, and how?" questions. Visualization techniques are introduced to aid analysis understanding. Consistency checks are defined over the interactive portion of the framework. Implementation, performance and potential optimizations are described. Group and individual case studies help to validate framework effectiveness in practice. Contributions are summarized in light of the requirements for early RE analysis. Finally, limitations and future work are described.

Acknowledgments

The completion of my doctoral degree has been a long, but rewarding, journey. I am indebted to many individuals for their assistance and guidance over the past five years and beyond.

I have been fortunate enough to study in a very active and supportive research lab at the University of Toronto, where I have received invaluable support from many students, professors, post-docs, and visiting researchers in our Software Engineering Lab. Their continual support and feedback over the course of many practice talks and discussions has been essential in formulating my research agenda. Our association with professors and students at the Faculty of Information has provided me with an additional research prospective which has influenced the content of this thesis.

I am grateful for the support and feedback of the iStar Community, received over the course of many workshops and discussions. I had the opportunity to visit the HCI research lab at City University London under the supervision of Professor Neil Maiden, gaining experience in the use of intentional modeling in large, realistic projects.

I have taken several graduate courses which were critical in providing me with the needed background to complete this work. This includes Requirements Engineering taught by Professor Steve Easterbrook, Conceptual Modeling, taught by Professor John Mylopoulos, Empirical Software Engineering, taught by Professor Easterbrook, and Automated Verification, taught by Professor Marsha Chechik. Projects for these courses, guided by the instructors, have evolved into several of the chapters and sections in this work.

My PhD committee members – Professors Easterbrook and Mylopoulos– have provided me with years of invaluable ideas, having great influence on my research. I am very fortunate to have had the opportunity to discuss my work with them over the years.

Most notably, I am indebted to the immense support and guidance of my PhD supervisor, Professor Eric Yu. He has spent countless hours providing ideas and feedback during many discussions and as part of iteration over numerous papers. Both the content of my thesis and my approach to research and writing have evolved immensely under his guidance. I am very grateful for the opportunity to have been under his supervision, first as an undergraduate student, as a Masters Student, and finally while completing my PhD. I have been especially fortunate to travel widely as part of my graduate studies, with the support of Professor Yu, allowing me to interact with some of the best researchers in the world.

I have made some wonderful friends during my graduate career, many of whom have been part of the SE or FIS labs at some point. I have appreciated them as both a source of support for my academic activities and a welcome source of distraction.

Finally, my family has been incredibly supportive of my academic endeavors over the last decade. They have remained proud of my achievements; even if it meant that I had to move far away to work on things most of them don't really understand. I am especially grateful for the love and support of my Dad, Jim Horkoff, who never questioned my plans to do something a little unusual for a girl from Vulcan, and my Mom, Cindy Johnson, who proofread almost all of my papers over the years, including this thesis.

This work has received financial support from Bell University Labs, Ontario Graduate Scholarship, and the Ontario Graduate Scholarship in Science and Technology.

Table of Contents

List of Tables

List of Figures

List of Appendices

Chapter 1 Introduction

1.1 Iterative, Interactive Modeling

Creating graphical models of a system and its environment can provide essential aid for system understanding and abstraction as a part of system design or redesign. As software systems become more complex and increasingly embedded within our daily lives, understanding the structure of the system, and the role of software systems in its environment becomes more challenging. The functionality of many practical systems is too complex for any one individual to understand in detail. In order to deal with this complexity, those involved in the development and operation of software systems have turned to conceptual modeling as a means of abstraction, communication and convergent understanding. Models can depict the structure of the system itself (e.g., component diagrams, UML class diagrams), system processes (e.g., business process models, UML activity diagrams), system data (e.g., ERD, DFDs), or system functionality (e.g., UML use cases, activity diagrams).

Creating and improving models in a way which is useful to the development or operation of a software system is challenging. When constructing a model, it is challenging to know when a model is (sufficiently) accurate or complete. Standards of completeness and accuracy themselves depend on the intended use of the models (correct/complete enough for what?).

One of the fundamental utilities of system models is their ability to support domain analysis of some kind. Typically analysis performed over models consists of using the structure or contents of the model to answer some question over the domain, in other words, using the representation and abstraction within a model to make important information more accessible. For example, in an ERD, what entities are related to what other entities in what way? Are these relationships correct in respect to the world? Are they potentially problematic for implementation (many-to-many, for example)? In an activity diagram, what happens when a certain condition fails? Is this what should happen in order to satisfy stakeholder needs? Models which are missing key information, or which contain inaccuracies can produce analysis results which lead to misunderstandings or poor system decisions. However, other than by checking against the domain knowledge of the modelers, it can be difficult to detect these issues. Often, it is only by executing the model in some way, for example through some form of model checking (e.g., Clarke, Grumberg, & Peled, 2000), or through transformations to executable code (e.g., OMG Group, 2009), that modelers can understand the accuracy and completeness of their models, and subsequently iterate over models to improve their quality. Ideally, model iteration would continue until the contents of the model stabilizes, in other words, further iterations reveal a minimal amount of changes or are not cost-effective to pursue. This process of iteration can be said to be interactive, as modelers must use their own knowledge of the domain and assessment of the analysis results to improve the model. These issues call for methods, approaches, or tools to aid in model analysis, specifically those which promote model improvement through interactive iteration.

Challenges concerning model completeness and accuracy are especially apparent when creating models which describe what will be (new or modified systems and their functionality) as opposed to what currently exists (existing systems and their roles). In this case, there is no objective "truth" or existing reality to check models against, only the shared perception of the system-to-be. In fact, the characteristics that make "to-be" modeling challenging – exploring that which does not yet exist – is precisely what makes this type of modeling especially valuable.

Models which focus on the functionality or role of a new or modified system are often used in the Requirements Engineering (RE) process as a means to elicit and represent the required functionality for a software system. Example models used in RE include UML Use Cases (e.g., Dano, Briand, & Barbier, 1997), Soft System Models (e.g., Checkland, 2000), and Problem Frames (e.g., Jackson, 2001).

Models can be especially useful in the early stages of requirements analysis. Although research and methods focusing on how to best represent or analyze requirements is useful, such methods often assume the requirements are known (Yu, 1997). The early stages of a requirements analysis process focuses on understanding the domain and the (often conflicting) needs of the stakeholders enough to gain a high-level understanding of the required functionality for the system-to-be.

Modeling and analysis can be especially challenging due to a number of factors which are prevalent when determining early requirements. For example, domain information in early project stages is often incomplete, or it is not worthwhile to drill down into too much detail during the early stages of analysis, especially when analyzing complex systems with much detail. The success of the system-to-be often relies on a number of important non-functional success criteria, such as security, privacy, or user satisfaction. These factors can be difficult to quantify or formalize with the time and information available in the early project stages. Even if partial quantitative measures are available, they may be difficult to combine or compare in order to derive overall recommendations (e.g., time vs. cost, products sold vs. market share, etc.) Despite this incomplete and imprecise information, early project stages are often where key decisions concerning project scope or functionality are made. Typically such decisions are made implicitly, without documentation or recorded rationale justifying the tradeoffs made between critical stakeholder criterions. In order to gain sufficient knowledge of stakeholder criteria, to foster a sense of user buy-in for the new system, and to continually validate collected domain knowledge, the involvement of key stakeholders is important for the early requirements process. In the context of early RE, with incomplete, imprecise information and a strong need for stakeholder involvement, it is especially important to aid users in an interactive process of model iteration, leading to relatively stable and complete models.

A particular type of model focusing on stakeholder goals has been proposed and used as a tool for system understanding and analysis in RE. Example goal modeling frameworks include the NFR Framework (Chung, Nixon, Yu, & Mylopoulos, 2000), GBRAM (Anton & Potts, 1998), KAOS (Dardenne, van Lamsweerde, & Fickas, 1993), i* (Yu, 1997), and AGORA (Kaiya, Horai, & Saeki, 2002), while example applications of this type of modeling in practice have been described by Maiden, Jones, & Manning, (2004) and van Lamsweerde, (2004). These models capture goals related to system functionality, including relationships between goals such as decomposition, conflict, or contribution. Part of the utility of goal models lies in their ability to include and compare the effects of alternative design or functionality in the model, aiding decision making. It has been suggested that such models are particularly suitable for elicitation and analysis in early RE as they can show the underlying motivations for systems, capture non-functional success criteria, support representation of qualitative success criteria, and show the effects of high-level design alternatives. Such models can be especially appropriate for early RE analysis when expanded to explicitly include systems, system stakeholders, and their interdependencies (Yu, 1997). We call this type of model, expanded to consider stakeholder and system agents, agent-goal models. Example agent-goal model frameworks include i*, GRL (Amyot, 2003) and Tropos (Bresciani, Perini, Giorgini, Giunchiglia, & Mylopoulos, 2004).

Like other models, one of the primary benefits of goal models is the ability to facilitate analysis, to use the information captured in the models to answer useful questions concerning the domain. The structure of goal models makes them especially amenable to an analysis of system objectives, when compared with other types of models used in requirements analysis. As goal models contain links describing the relations between goals (e.g., decomposition and help), it is natural to trace these links from the selection of a particular goal to other goals along the path of links, propagating the "satisfaction" of goals onto other goals (Chung et al., 2000). This type of analysis allows modelers to pose useful questions over the model. For example, what are the effects of implementing a certain set of functionality on stakeholder's goals? Or, is it possible for a certain set of goals to be satisfied, given the functional alternatives captured in the model? A subset of work devoted to goal model analysis focuses on answering such questions by applying systematic propagation of goal satisfaction (or conversely, "denial") to their models (e.g., Chung, Nixon, E. Yu, & Mylopoulos, 2000, Amyot et al., 2010, Giorgini, Mylopoulos, & Sebastiani, 2005).

The contents and structure of goal models facilitates other types of analysis methods, answering a variety of analysis questions. For example, the application of metrics over model structure can check properties such as predictability (e.g., Franch, 2006), application of planning techniques can answer questions about task sequences or plans (e.g., Asnar et al., 2007), simulations can test how operation affects goal achievement (e.g., Gans et al., 2002), model checking can be used to check for formal properties such as eventual goal achievement (e.g., Fuxman, Pistore, Mylopoulos, & Traverso, 2001).

When capturing socio-technical domain, goal models typically become large quickly. It becomes difficult for modelers to trace and understand the affects of specific functional alternatives in the model. Many of the procedures introduced for goal model analysis focus on automated reasoning, placing more value in the results of the analysis than in the interactive process of analyzing and exploring the model. Given the complexity of goal models and the demand on user time, automated reasoning can be appealing. However, because of the relatively incomplete nature of goal models, the inexpressiveness of goal model frameworks, and the focus on social domain aspects, analysis results over agent-goal models should be treated as a heuristic, possibly containing inaccuracies depending on the quality of the model and the modeler's knowledge of the domain. Many of these procedures require the modelers to add additional detailed information to the model, such as temporal ordering, or quantitative metrics, making the underlying modeling framework more expressive. However, the information required is often the same information that is difficult to elicit in the early stages of requirements analysis. Focusing on such details may distract analysts and stakeholders from capturing and sufficiently understanding the “big picture”.

Most existing goal model analysis procedures do not focus on the role the modeler plays as part of the analysis. Ideal analysis procedures for early RE agent-goal models would consider the social, qualitative and incomplete nature of such models, would facilitate iteration and domain exploration, and would support stakeholder interaction.

Ideal procedures for early RE modeling would prompt iteration over domain knowledge, increasing the likelihood of discovering objectives, problems, and alternative remedies in the domain, moving the model towards relative completeness and accuracy. We are interested in methods which allow interaction, receiving frequent input from stakeholders, but which can be enhanced by tool support in order to allow modelers to analyze large and complex models. Much of this understanding gained through early RE analysis is of a social nature, and difficult or impractical to completely express formally. As such, model contents can be subjective, and methods are needed in order to help create consistency in the interpretation of domain information captured in the models. As models are intended to aid decision making in early RE, it is important to capture the rationale behind such decisions, supplementing the model and analysis results with user judgments. Methods to guide use of the modeling and analysis notations are needed.

This thesis addresses these needs by creating a framework for iterative, interactive analysis of agent-goal models in early requirements engineering. The main objectives of this work can be summarized in the following thesis statement:

> This work aims to provide methods, algorithms, and tools to facilitate interactive, iterative analysis of early agent-goal requirements models. This type of analysis aims to provide analysis power, improve model quality, increase domain knowledge, encourage the involvement of stakeholders, and capture judgments over important decisions. The framework will provide scalable methods which are usable in practice. We claim that such contributions will ultimately lead to the development of more effective software systems.

We motivate the focus on iterative, interactive analysis of agent-goal models for Early Requirements with a more detailed example concerning a youth counseling organization.

1.1.1 Motivating Example: Youth Counseling Organization

Consider an example organization: a not-for-profit organization that focuses on counseling for youth over the phone, but must now expand their ability to provide counseling via the Internet. Online counseling could be viewed by multiple individuals, and may provide a comforting distance which would encourage youth to ask for help. However, in providing counseling online, counselors lose cues involved in personal contact, such as body language or tone. Furthermore, there are concerns with confidentiality, protection from predators, public scrutiny over advice, and liability over misinterpreted guidance. The organization must choose amongst multiple technical options to expand their internet counseling service, including a modification of their existing anonymous question and answer system, discussion boards, wikis, text messaging, blogs, etc.

A high-level understanding of the organization, the existing system, and its current users is needed. The organization must have the self-reflective information available to make decisions over the future of online counseling and the resulting software system.

Methods described in previous work (Yu, 1997), can be applied to understand the domain, including stakeholders, system, goals, contributions, and dependencies. The results of which may be one or more large agent-goal models. See Figure 1 for a high-level overview of an example agent-goal model created for this domain. Figure 2 contains a simplified representation of part of the domain for better understanding. In this model, the Counseling Organization must choose between several forms of providing online counseling. Their choices affect not only their goals, but also the goals of the Counselors and the Kids and Youth. The model contains three actors: the Organization (top), Kids and Youth (bottom left), and Counselors (bottom right). The Organization, an agent, wants to achieve several softgoals, including Helping Kids, Increasing Funds, and providing High Quality Counseling. These goals are difficult to precisely define, yet are critical to the organization. The Organization has the "hard" goal of Providing Online Counseling Services and explores two alternative tasks for this goal: Use Text Messaging and Use Cyber Café/Portal/Chat Room. These alternatives contribute positively or negatively by various degrees to the Organization's goals, which in turn contribute to each other. For example, Use Text Message hurts Immediacy which helps High Quality Counseling.

The Organization depends on the Counselors to provide the alternative counseling services and for many of its softgoals, for example, High Quality Counseling. Kids and Youth depend on the Organization to provide various counseling services, such as Cyber Café/Portal/Chat Room. Both the Counselors and Kids have their own goals to achieve, also receiving contributions from the counseling alternatives. Although the internal goals of each actor may be similar, each actor is autonomous: High Quality Counseling may mean something different for the Counselor than for the Organization.

Examining the example model raises several questions: Which counseling alternative is the most effective, and for whom? Are there alternatives which could achieve each actor's goals? If not, why not? What important information is missing from the model? Is the domain captured effectively? Generally, how can such an organization explore and evaluate options for online counseling, balancing the needs of multiple parties, while dealing with the complexity of the model and domain?

Although some questions may be answered by studying the model, tracing effects consistently without guidance quickly becomes complex. The model in Figure 2 is a simplified version of a larger model, such as the one in Figure 1. If tracing the effects of alterative functionality is difficult in the second model, it is especially challenging in the first model. This example demonstrates the need for systematic analysis procedures which helps the modeler to trace effects in the model in order to answer domain questions and evaluate alternatives. However, such procedures must account for the inherent incompleteness and inaccuracy of early requirements models.

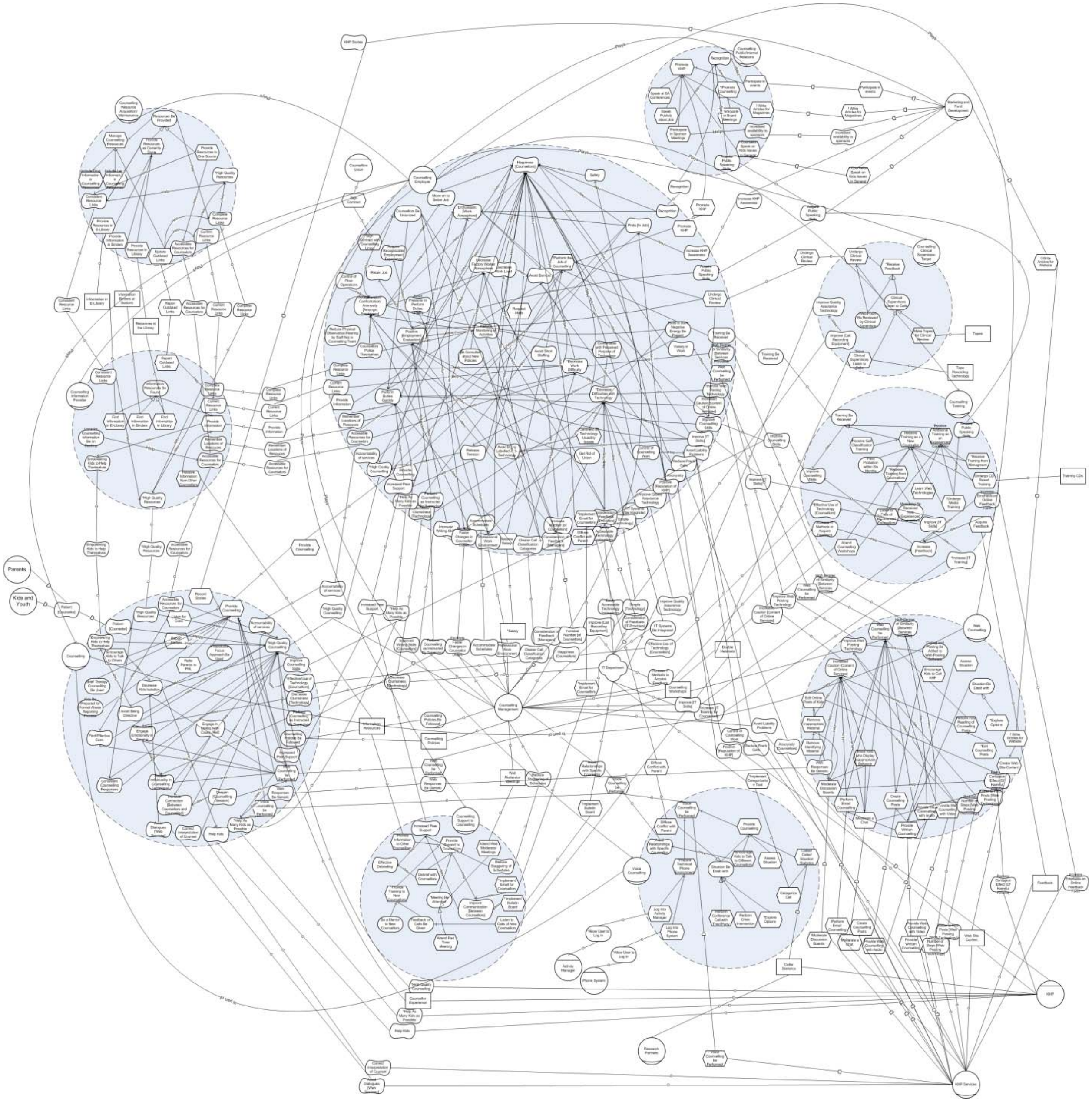


Figure 1: High-Level Overview of an i* Model Describing the Current Counseling Situation for the Social Service Organization Example

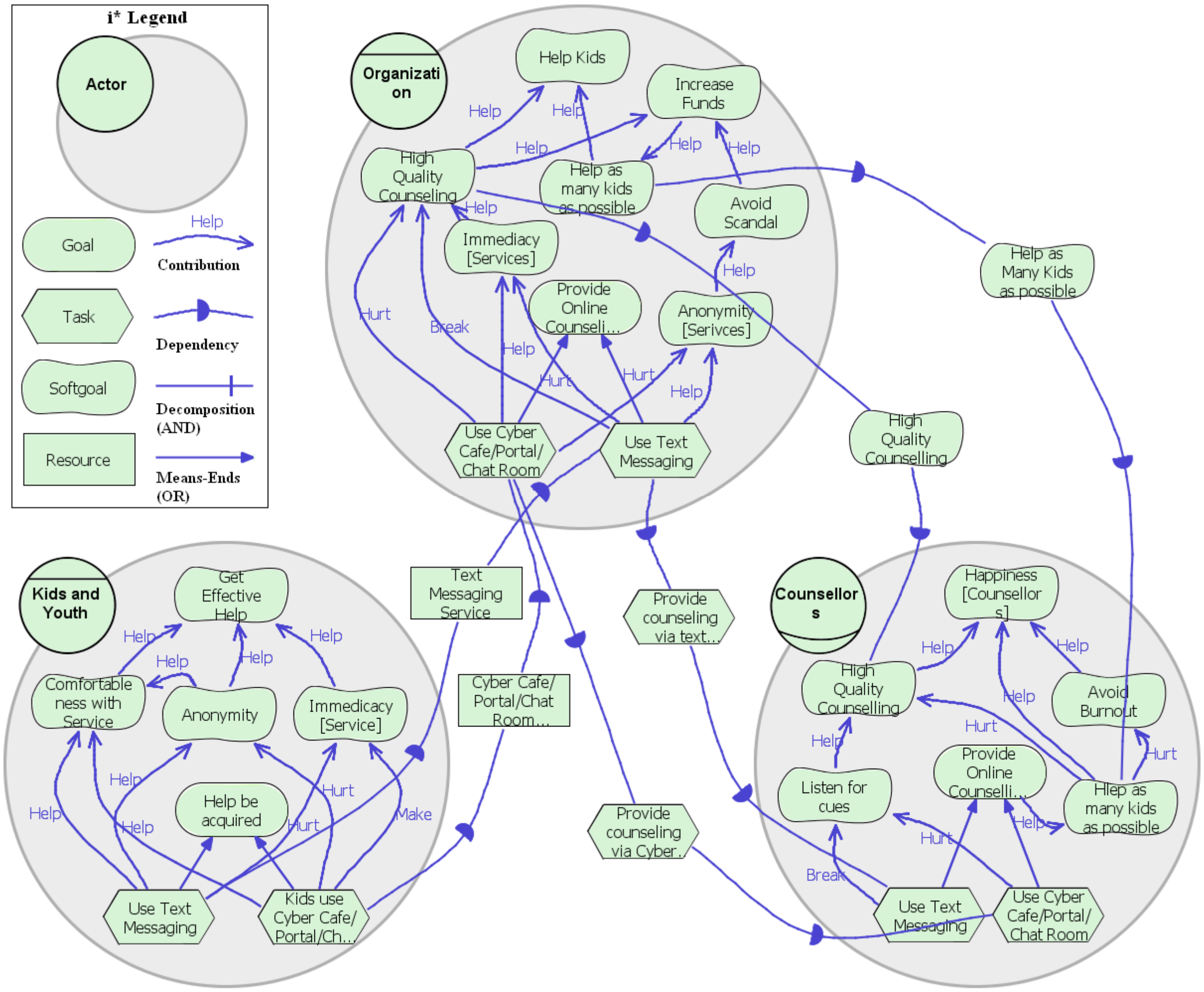


Figure 2: i* Model Representing Simplified Relationships and Alternatives for Online Counseling

1.2 Challenges in Agent-Goal Model Analysis for Early Requirements Engineering

We have identified challenges specific to early requirements analysis such as the importance of non-functional criteria, and the struggle for sufficient model accuracy and completeness. These challenges play an important role in the operation and effectiveness of analysis procedures used for agent-goal models in early requirements analysis. In this section, we outline several of the challenges in analyzing agent-goal models in early RE.

Model Complexity. As can be seen in Figure 2, agent-goal models can become too complex to be analyzed by hand. One way to help alleviate the complexity of analysis over goal models is to automate the analysis process. Some level of automation is needed to support analysis of the contents and structure of the model. Finding the appropriate level of support can be challenging. Too much automation fails to account for inherent model inexpressiveness and incompleteness in early RE, and may limit the role of the modeler or stakeholders. Too little automation can lead to a process which is too tedious or time-consuming to realistically complete, and may produce results which are inconsistent and difficult to reinterpret at a later time.

It can also be challenging to understand the results of analysis over a complex model. Methods which help model or analysis comprehension would help such analysis to be more accessible to stakeholders with limited time.

Model Completeness. Because of the high-level, social nature of early RE models, it can be argued that models are never complete in the same way as models used for other purposes. In other modeling contexts, there are often clear criteria for model inclusion. For example, if making an ERD of an existing system, the entities involved in the system of focus are finite. When modeling early requirements, models depict only a part of a complicated interconnected web of systems and people. There are always more stakeholders, more interacting systems, more goals, dependencies and contributions that can be added to the models. The difficulty in doing such modeling is in knowing where to stop modeling, and knowing if the amount of information collected is sufficient to support useful analysis and understanding of the domain.

Model Accuracy. Issues for model accuracy are similar to those for model completeness. When representing abstract social needs and interactions, it is difficult to produce a model which is completely accurate. In fact, it can be argued from a constructivist viewpoint that there is no one “correct” model, but only models which sufficiently capture the collective viewpoint of the modelers concerning the domain. The challenge in early RE modeling is to improve the accuracy of models to a stable point, and to create a model which is accurate enough for useful analysis and understanding.

Domain Understanding. One of the primary aims of modeling and analysis in early RE is increasing understanding of the domain, in order to better understand the requirements of a new or improved system which effectively balances needs in the domain. It can be especially challenging to understand the domain early in the requirements process, when trying to understand a large, complex, socio-technical organization. Often stakeholders will wish to focus on technical details or requirements without getting into the motivations behind those requirements, or into the conflicting goals which may impede the success of technical choices in practice. The challenge is to use early RE models and analysis to improve domain understanding to a sufficient level in order to specify an effective technical intervention, knowing that a full knowledge of all details in the domain is not feasible given project time constraints.

Model Interpretation. The inexpressive and qualitative nature of goal models can lead to differing interpretations amongst modelers. Inconsistencies in model understanding can also derive from the nature of goal model syntax. Goal model constructs in some frameworks are designed to support expressiveness by leaving concepts open to a certain degree of interpretation. For example, the means-ends link in i* as defined by Yu, (1997), is meant to represent alternative means for achieving a goal, with goals able to be achieved in more than one way. This definition has left the "or" concept open to an inclusive or exclusive logical interpretation. Methods which help modelers increase their consensus on the meanings and interpretations of constructs in their domain models would help to increase knowledge sharing in the early requirements process.

Model Flexibility. Goal models were developed specifically to offer a level of flexibility and inexpressiveness in order to facilitate the explicit consideration of high-level, non-functional and social requirements (e.g., customer satisfaction, company branding). Such requirements are difficult to quantify and formalize, but should be captured and considered in the early stages of analysis. Even if requirements can be assigned clear-cut measures in the early stages, it is challenging to determine how measures in different scales can be combined in an intuitive and accurate manner. Once key decisions have been made, and the scope of the system has been narrowed or sufficiently partitioned, critical system qualities can be quantified or formalized for later verification. However, models for early RE must support reasoning over early, inexpressive, ambiguous representations of these critical system requirements.

Decision Rationale. Typically system decisions made in early RE are made implicitly, possibly via group consensus, and often go without being documented. The use of goal models and various analysis procedures can capture some of the information used to make key early decisions. However, even with the model and analysis results, it may not be clear why certain decisions were made, including the judgments made over contentious areas in the model and the assumptions which supported stakeholder choices. It is challenging to know what information associated with a goal-model aided decision process should be recorded. Too much information makes the models and the modeling process overly complex, too little information leaves stakeholders wondering about their own or other's choices.

Stakeholder Involvement. It is important for stakeholders to be an active part of the early RE process. Stakeholders must be involved to provide information on their needs and interactions, and to validate the information already collected. Often, stakeholders' busy schedules make it challenging to access their time. Depending on the domain, system users may be reluctant to support impending changes, for technical or political reasons. The complexity of goal models in practice means that stakeholders often have a difficult time understanding models, their purpose, or how to usefully interpret them for analysis. The inherent incomplete and inaccurate nature of goal models means that it is especially important to encourage continual iteration as a means of stakeholder validation. In addition to improving the accuracy and completeness of the models, stakeholder participation in the early system analysis can induce "buy-in" or a sense of ownership in project goals and planned changes. Methods are needed to involve the stakeholder as much as possible in the modeling and analysis process. Given the scarceness of stakeholder time, approaches to goal modeling and analysis should have a relatively low learning curve, and should have motivations and functionality which are reasonably transparent to users.

Analysis Power. Although the process of creating an agent-goal model can be useful for understanding and consensus even without analysis, in order to increase the payoff for time invested, such models should support as much analysis as possible. In other words, modelers should be able to use the models to answer several types of useful domain questions. Although the structure of goal models allows for the application of multiple procedures which may answer different questions, most of these procedures require the addition of specific formal or quantitative information to the goal models (e.g., Letier & van Lamsweerde, 2004). Here, there is a tradeoff between including more specific information into the goal model in order to facilitate different types of analysis (e.g., probabilistic, temporal) and the difficulty in finding this type of information in early RE, including the time required to encode such information in the model. The expressiveness and completeness of the goal model plays a role in the accuracy and trustworthiness of the resulting answers. It is challenging to know how much trust modelers and stakeholders should place in goal model analysis results. If too must trust is given, model inaccuracies or misinterpretations could lead to poor decisions, but too little trust could fail to take advantage of the utility of goal models and goal model analysis for domain understanding and exploration.

Procedure Usability. Although many existing approaches for agent-goal model analysis have been introduced, very little work focuses on the practical usability of these procedures. It is not clear whether these procedures are usable, or what expertise may be needed to apply them effectively. Several approaches introduce complexities either by expanding goal model syntax (e.g., Gans, Jarke, Kethers, & Lakemeyer, 2003) or by introducing a complex analysis methodology (e.g., Asnar & Giorgini, 2006). Procedures which explicitly aim for simplicity and usability are needed.

Procedure Selection. Given the variety of available procedures for goal models, and the wide range of analysis questions they facilitate, how can potential goal model users choose analysis procedures that are right for their needs? What procedures would they choose if they were performing analysis in an early RE content? Understanding benefits, capabilities, and costs within the range of possible goal model analysis techniques is challenging.

We summarize the challenges in agent-goal model analysis for early RE using a simple goal model in Figure 3. Challenges have been reworded slightly in their representation as goals (e.g., Stakeholder Involvement is now Involve Stakeholders). This model includes some of the perceived synergies and conflicts amongst the challenges. For example, capturing stakeholder decisions and rationale would help to involve stakeholders, but may make the resulting models and procedures more complicated (and less usable).

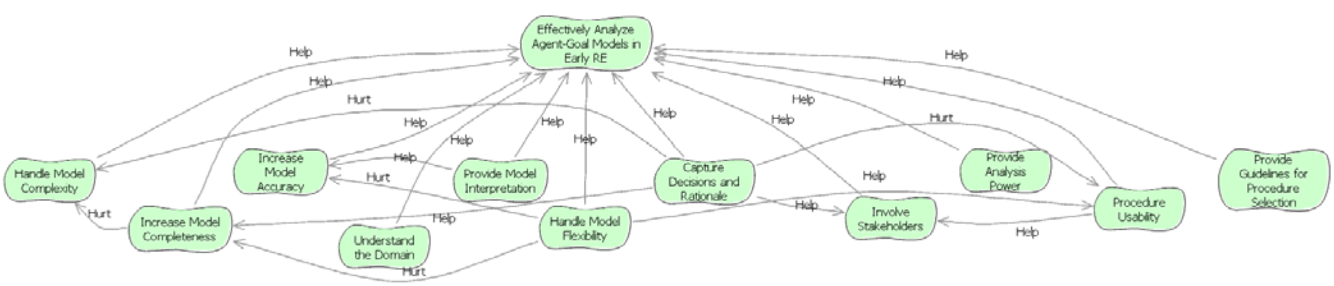


Figure 3: Summary of Challenges for Agent-Goal Model Analysis

1.3 Existing Approaches to Goal Model Analysis

We can identify a number of procedures which analyze the satisfaction or "denial" of goals in a model. These procedures start with initial values assigned to the model, reflecting an alternative or question, and then use model links to propagate values either forward (in the direction of the link), (Amyot et al., 2010, Chung et al., 2000, Giorgini et al., 2005, Letier & van Lamsweerde, 2004, Tanabe et al., 2008, Maiden, Lockerbie, Randall, Jones, & Bush, 2007, or backward, (Giorgini, Mylopoulos, & Sebastiani, 2004, Letier & van Lamsweerde, 2004, Giorgini et al., 2005). These procedures can answer questions like "What is the effect of this alternative?" (forward) or "Can certain goals be satisfied? If so, what alternative in the model would satisfy these goals?" (backward).

Some satisfaction analysis procedures present results in terms of qualitative labels representing satisfaction or denial, typically using: (sufficiently) satisfied, partially satisfied, (sometimes) conflict, none/unknown, partially denied, and denied, (Amyot et al., 2010, Chung et al., 2000, Giorgini et al., 2005). Several procedures offer quantitative analysis, using numbers to represent the probability of a goal being satisfied or denied (Letier & van Lamsweerde, 2004, Giorgini et al., 2005, Tanabe et al., 2008), or to represent the degree of satisfaction/denial (Amyot et al., 2010). Other procedures produce binary results, where goals have only one of two values, typically satisfied or not. For example, Maiden et al., (2007) analyzes in terms of compliance, whether an argument can be made to justify the satisfaction of tasks and resources based on existing requirements.

One of the primary distinguishing features between these approaches is their means of resolving multiple incoming values for goals. Goal models often include contribution links representing positive and negative consequences of various degrees. A goal could receive several different types of contributions at once, positive and/or negative of various strengths. Some procedures deal with such situations by separating negative and positive evidence, making it unnecessary to resolve conflicts (Giorgini et al., 2004, Giorgini et al., 2005). Other procedures make use of predefined qualitative or quantitative rules to combine multiple values (Amyot et al., 2010, Letier & van Lamsweerde, 2004). Further procedures are interactive, using human intervention based on domain knowledge to resolve partial or conflicting evidence (Chung et al., 2000).

Further approaches use metrics over constructs in the model to measure qualities over the domain such as security, vulnerability, and efficiency, (e.g., Franch & Maiden, 2003, Franch, 2006). These procedures can answer questions like "How secure is the system represented by the model?" or "How risky is a particular alternative for a particular stakeholder?" For example, Franch, Grau, & Quer (2004) introduce the means to calculate global or local metrics over i* Strategic Dependency (SD) models using classifications and weights of actors and dependencies in an SD model, then expand this approach (Franch, 2006) to work over i* Strategic Rationale models, developing a framework which allows for qualitative or quantitative, automated or interactive metric calculation.

Methods have applied AI-type planning to find satisfactory sequences of actions or design alternatives in goal models (e.g., Asnar et al., 2007, Bryl et al., 2006a, Gans et al., 2002). These procedures can be used to answer questions such as "What actions must be taken to satisfy goals?" or "What are the best plans of action according to certain criteria?" For example, Bryl et al., (2006) aim to find satisfactory delegations (assignment of dependencies) in a social network represented via goal model by iteratively finding plans within the model that fully satisfy all actors, and then evaluating the plans in terms of cost, similar to the metrics used in (Franch, 2006).

Several approaches have added temporal information to goal models to allow for simulation over the network represented by model constructs (e.g., Gans et al., 2002, Wang & Lespérance, 2001, Gans et al., 2003a). In these approaches, a particular scenario is simulated, and the results are checked for interesting or unexpected properties. These procedures can answer questions like "What happens when a particular alternative is selected?"

Further approaches provide ways to perform checks over the models supplemented with additional information, allowing users to ask questions like "Is it possible to achieve a particular goal?" or "Is the model consistent?" For example, Fuxman et al. (Fuxman, Liu, Pistore, Roveri, & Mylopoulos, 2003; Fuxman, Pistore, Mylopoulos, & Traverso, 2001) convert i* models to Formal Tropos, supplementing the models with first order linear-time temporal logic statements to represent desired constraints, and a model checker is used to validate properties and check for consistency. Although the checks are automatic, an iterative process of manually defining the bounds of the model checker is often required.

Other approaches focus on methods for model construction instead of analysis (e.g., (Grau, Franch, Mayol, Ayala, & Cares, 2005; Grau, Franch, & Ávila, 2006; Letier & van Lamsweerde, 2002). These methods help modelers, especially those new to the notation, to create models. Such methods include the Process Reengineering i* Method (PRiM), where i* models are constructed in a systematic manner, separating operational from intentional content and using complementary artifacts such as context models and Data Flow Diagrams (Grau, Franch, & Ávila, 2006).

1.3.1 Limitations of Existing Approaches

Section 1.2 outlined several challenges in agent-goal model analysis for early Requirements Engineering. Existing work typically does not explicitly define early RE challenges, with most work focusing mainly on the analytical power offered by their procedures. Many procedures for goal model analysis are aimed at addressing problems outside of the scope of early RE, for example to simulate detailed system operation (Gans et al., 2005). In this section, we discuss the contributions of existing work as they relate to the challenges of early RE analysis.

Model Complexity. Some level of automation is needed to support analysis over complex goal models. The majority of existing procedures for goal model analysis are fully automated, taking some input provided by the user and using the contents of the model to automatically produce analysis results. However, given the incomplete and inaccuracy of early RE models, it is hard for users – analysts and stakeholders – to trust the results of fully automated analysis. Depending on the complexity and the transparency of the analysis procedure, and the complexity of the model it may be difficult to understand how results are achieved, making validation of results challenging. For these reasons, we argue that fully automated analysis is not ideal for analysis in the early stages of RE.

Some analysis procedures have introduced interactive components, allowing the modelers to intervene at various points in the procedure (e.g., Asnar, Bryl, & Giorgini, 2007; Franch, 2006; Maiden, Lockerbie, Randall, Jones, & Bush, 2007). For example, in analysis in the NFR procedure, analyst intervention is used to promote or demote partial evidence, or to decide if the evidence is conflicting. Although it is useful allowing the modeler to intervene in the modeling process with their domain knowledge, the restrictions on user intervention, specifically that all values must be promoted, can be limiting and lead to loss of information.

Existing approaches have not focused on aiding the comprehension of analysis results over complex models. If stakeholders are to play an active role in the modeling and analysis process, methods to aid analysis comprehension are needed.

Model Completeness. To our knowledge, goal model analysis procedures do not address issues of model completeness. In fact, most procedures proceed with the assumption that the models being analyzed are complete and accurate. Existing procedures focus on analytical power and not on determining the gaps in knowledge captured by the model. Although any analysis procedure could potentially reveal important missing information when the results are examined by stakeholders, finding omissions is especially hard when analyzing the "to-be" situation. When analyzing early RE models it may be especially helpful for modelers to reexamine and question the completeness of the model, particularly fragments which have a high degree of uncertainty. Automatic analysis procedures may not encourage such reexamination. Finding and examining these areas in an ad-hoc way may be challenging for modelers, especially if they have already spent much time creating the model and have reached some agreement on its contents. Methods are needed as part of model analysis to aid modelers in iterating over important model fragments, finding potential omissions, and improving the quality of the model.

Approaches for model construction help modelers to achieve completeness by providing suggested steps for model construction and by providing checks against other types of models (e.g., Grau et al., 2006). This can be especially helpful when first constructing a model. Due to the complex nature of agent-goal models, not all creation steps can be easily dictated, some creativity is required to create a complete model. As agent-goal models have an intention-centered focus, checks against non-intentional models can only go so far in improving the completeness of the model. Further approaches are needed to increase confidence in the completeness of goal-oriented models, helping to reach model stability.

Model Correctness. Like analyzing the completeness of a model, the results of any analysis procedure, when examined by the modelers, could reveal inaccurate pieces of the model. However, to our knowledge, existing procedures focus on analyzing the domain as represented by the model, and not on the correctness of the model itself. Analysis procedures have potential to be used as a means to check model correctness, especially as basic "sanity" tests when models are first created. Use of goal analysis procedures for this purpose requires user guidance, including the types of analysis questions to ask over newly finished models.

Similar to model completeness, it is useful for modelers to continually reexamine key areas of the model, in this case to find model inaccuracies. Such a process may not be well supported by fully automated analysis. There is a need for procedures which find a balance between automation and intervention more appropriate for goal models used in early RE.

Model Interpretation. By defining analysis methods over goal models, most procedures enforce a more specific interpretation over model constructs. For instance, returning to the i* example from Section 1.2, all procedures which work over i* models define a specific way of interpreting the means-ends link, either as an inclusive or exclusive OR relation. However, most procedures do not focus on consistent interpretation as a goal of their methods. When imposing their interpretation over the model syntax, several analysis procedures make different assumptions concerning the meaning of goal model constructs. For example, some procedures treat evidence through contribution links as necessary in order to achieve a softgoal, while others look at it as an accumulation of values. Current work has not explicitly examined whether or not inconsistent interpretation is an issue in practice, and whether or not systematic analysis helps users make more consistent interpretations of the model. Studies which explicitly examine these issues guide the development of appropriate analysis procedures for early RE Goal Models.

Domain Understanding. Any procedure can help to understand the domain by providing analysis results over a domain model. However, existing procedures do not explicitly aim to increase or improve domain understanding, but instead focus on answering specific questions; often selecting the best amongst a series of technical alternatives, without ensuring the selection criteria is sufficiently accurate or complete.

Model Flexibility. Several of the procedures support reasoning over flexible, inexpressive models via the use of simple, qualitative labels (e.g., Amyot et al., 2010; Chung, Nixon, E. Yu, & Mylopoulos, 2000; Horkoff, 2006). Such labels can be applied and propagated using a mix of automated rules and stakeholder judgment without forcing users to formalize or quantify high-level model concepts. However, other procedures use a quantitative interpretation over these informal concepts, assuming that the numbers are meaningful, i.e. customer satisfaction = 0.7 means that this goal is satisfied on a scale of 7/10 (e.g., Amyot et al., 2010; Giorgini et al., 2004; Kaiya et al., 2002). Yet other procedures require that the model have a precise formal or quantitative meaning before undergoing analysis (e.g., Bryl et al., 2009b; Fuxman et al., 2003; Letier & van Lamsweerde, 2004). Several approaches require the addition of specific information such as cost, timing, or probability of occurrence in order to evaluate a model (e.g., Gans et al., 2002; Giorgini et al., 2004; Letier & van Lamsweerde, 2002; van Lamsweerde, 2009). Requiring formal, quantitative, or detailed representations for high-level social concepts limits the flexibility and usability of these approaches for early RE.

Capturing Rationale for Decisions. Analysis results can be used as a form of rationale for decisions. However, this means that users need to be able to easily compare results, in order to understand why a particular analysis scenario was thought to be preferable to another. If procedures move away from full automation, in order to involve the users and make use of their domain knowledge, decisions made over the model should be captured and stored in some way. Modelers should be able to return to these decisions in order to remember why they were made, and change them if desired. Current interactive goal model analysis procedures do not provide support for the organization of or iteration over model decisions. It would be especially helpful to allow modelers to capture free-form rationale for decisions over the model, attached to the model in some way. Although work by Maiden, Lockerbie, Randall, Jones, & Bush, (2007) supports the storage and management of satisfaction arguments over model decisions, their approach applies these arguments over limited structures in the model, and does not emphasize modification of or iteration over arguments.

The Role of the Stakeholder. Stakeholders should be an active part of the early RE process to provide and validate domain information. Current analysis procedures generally do not focus on the role of the stakeholder in analysis. Most procedures exclude the modeler from the analysis process, only taking input at the beginning of the process, to frame the query over the model, and providing analysis results output at the end. Some procedures allow for expert intervention at certain points (e.g., Asnar et al., 2007; Franch, 2006; Maiden et al., 2007). Typically in these procedures the participation of "experts" is seen as a necessary step in order to enhance the model or analysis with domain knowledge, but it is not explicitly encouraged as a means to involve the user. Encouraging user involvement in goal model analysis in a clear structured way allows for a higher level of user input, encouragimg iteration over the correctness and completeness of the model, and increasing chances of achieving stakeholder buy-in in the new system.

Analysis Power. Existing procedures support a wide range of analysis questions over models. However, most procedures require specific quantitative or formal information to be added to the model, or produce results automatically over high-level models. Procedures are needed which provide a range of analysis capabilities, but which keep in mind the available information, the nature of early RE models, and the role of the user. Analysis results over models which may be incomplete or inaccurate to a certain degree should be given appropriate weight in domain understanding and decision making, as part of a methodology for early RE exploration.

Procedure Selection. Although existing techniques often review related procedures for goal model analysis, few aim to specifically guide technique selection based on criteria in or characteristics of the goal model domain.

1.4 Iterative, Interactive Agent-Goal Model Analysis Framework

In order to address the challenges in agent-goal model analysis for early Requirements Engineering and the limitations of previous work applicable to this area, we provide a framework to support interactive, iterative analysis over agent-goal models used in early RE. As a first step, we review existing procedures in detail, summarizing their analysis capabilities including the specific information (e.g., quantitative measure, temporal ordering) they need to perform analysis. This information is used to make recommendations for procedure selection depending on characteristics in the domain. Next, we examine the characteristics of the early requirements process in order to derive a list of requirements for early RE analysis. We use these requirements to evaluate the suitability of existing goal model analysis procedures for early RE, selecting existing procedures for inclusion in this framework. As a next step, we perform a detailed analysis of methods for forward satisfaction propagation. Differing interpretations of goal model syntax lead to differing propagation rules for model evidence. The purpose of this analysis is to understand to what degree these different propagation rules affect analysis results, and to make recommendations concerning the heuristic role of goal model analysis in early RE.

After reviewing and analyzing existing goal model analysis techniques, we introduce the specific elements of our framework. We begin with a review of i* syntax, including common syntax variations in academic and student use. A more formal description of i* is provided, attempting to balance the need for precision with syntax flexibility in order to facilitate reasonable variation. Next, we describe analysis procedures over the i* framework. The framework itself will allow for multiple types of qualitative analysis, manage alternative evaluations over a model, manage interactive results, provide visual interventions to aid analysis comprehension, and will include methodologies to guide users in modeling and analysis. The framework is evaluated in several ways. Forward analysis has been applied to several examples and to the large counseling case study described in Section 1.1. A small experiment has been conducted to test some of the perceived benefits of forward analysis. The implementation of forward and backward analysis has been tested in ten individual case studies and in one group academic setting. A follow-up of the individual case studies has been used to test visual comprehension aids. Components of the framework are described in more detail in the rest of this section.

Goal Model Analysis Review and Selection Criteria. Although the variety of methods for goal model analysis is encouraging from a research perspective, from the perspective of practitioners or potential goal model users the diversity of analysis procedures available can be confusing, thus limiting their adoption. Before developing our goal analysis framework, we address two objectives: survey available approaches for goal model analysis in order to understand existing approaches, and provide initial guidelines for procedure selection. As part of this survey, we aim to answer: What methods are available? What types of analysis questions can these methods answer? What goal model constructs or notations do the procedures support? What domain information is needed in order to use the methods? What are some of the potential benefits of goal model analysis in the requirements process? Which available methods can be applied to achieve which kinds of usage objectives? How can we use this information to advise on selection? Are existing analysis approaches appropriate for early RE? Can we use existing analysis approaches as part of this work? How can we use the shortcomings of existing frameworks to direct the development of our early RE analysis framework? Details concerning the survey and selection guidelines have appeared as part of Horkoff & E. Yu (2011a).

Forward Satisfaction Detailed Comparison. A comparison has been made between several forward satisfaction analysis procedures for goal models. Three available tools implementing seven similar goal satisfaction analysis procedures are used to analyze three sample goal models. Results are reported and compared. The purpose of this comparison is to understand the ways in which procedural design choices affect analysis results, and how differences in analysis results could lead to different recommendations in the use of goal models for early requirements analysis. Results are used to make recommendations on the use of satisfaction analysis techniques for goal models as part of the interactive iterative framework, advocating their use as heuristics guiding domain exploration and decision making, and emphasizing benefits beyond analytical power. A version of this comparison and analysis will appear in Horkoff & E. Yu (2011b).

Reflective Analysis and Definition of the i* Agent-Goal Modeling Framework. We have selected the i* Framework for use as an example agent-goal model thesis, due to its ability to balance between expressiveness and flexibility, its popularity in research, and its use as part of our selected component procedure from Horkoff (2006). As the i* Framework aimed to define a framework flexible enough to facilitate modeling of early requirements, the description of the Framework was left open to a certain degree of interpretation and adaptation. When i* is used in practice, its syntax is often modified, either deliberately or accidentally. We survey common variations made by researchers or students, analyzing the motivations behind variations and classifying the variations as permissible shortcuts (warnings) or incorrect syntax use (errors). Results from this survey are used to create a more formal definition of i*, allowing for as many permissible deviations as possible, while still making concepts more precise. This definition is used in the analysis procedures and features in the remainder of the work. Details of the survey of i* syntax have appeared in (Horkoff, Elahi, Abdulhadi, & E. Yu, 2008), while details of the i* formalism have appeared in (Horkoff & E. Yu, 2010a).

Forward Analysis Procedure. An interactive, qualitative forward evaluation for i* models has been developed as part of Horkoff (2006). We chose to incorporate this procedure in the analysis framework introduced in this thesis. As part of the adoption of the existing analysis procedure, we re-describe it using more formal concepts, re-examine procedure termination, and provide further analysis examples. The procedure starts with an analysis question of the general form "How effective would a proposed solution be in meeting the desired goals?" The analysis makes use of a set of qualitative evaluation labels, assigned to intentions to express their degree of satisfaction or denial. The procedure propagates initial values iteratively from contributing intentions to recipient intentions through model links using defined rules. The interactive nature of the procedure applies when human judgment, based on domain knowledge, is used to combine multiple conflicting or partial values to determine the satisfaction or denial of a softgoal. An assessment is made as to whether the alternative is satisfactory, stimulating further analysis and potential model refinement. Details concerning this procedure have appeared in (Horkoff & E. Yu, 2009a, 2009b, 2010b).

Backward Analysis Procedure. In addition to "What-if?" questions, it is useful to support "Is it possible?" questions. For example, "Is is possible for certain intention(s) in the model to be satisfied? And, if so, what alternative produces these results?". Answering these questions requires "backward" analysis, where desired values are placed on the model and the procedure works backwards (from recipient intentions to contributing intentions) to find alternatives in the model which produces these values. Work in (Giorgini, Mylopoulos, & Sebastiani, 2004) has implemented a fully-automated, two-value procedure for non-agent goal models using a SAT solver. We expand on this approach, adapting it to consider agent-oriented concepts, a single evaluation value for each intention, and the role of human intervention, producing an iterative, interactive procedure. Descriptions of the procedure can be found in (Horkoff & E. Yu, 2008, 2010a).

Analysis Visualization. The framework supports several visualization techniques over goal models to help the user initiate and understand analysis results. Specifically, we highlight model leaves and roots as suggested starting points for model analysis, highlight intentions involved in a human judgment, and highlight goals directly involved in conflict situations in backward analysis. These interventions have been described in (Horkoff & E. Yu, 2010c).

Judgment Inconsistencies. In order to encourage useful iteration over the models, and to highlight areas of interest over the model, we introduced checks for model judgment consistencies. Judgments are made as part of forward or backward analysis to resolve partial or conflicting evidence over contentious areas of the model. We check these judgments for consistency with the structure of the model, or between multiple judgments over a single intention. Inconsistent judgments are areas for potential model iteration or domain discussion and consensus building.

Suggested Methodology. Application of the framework is guided by providing suggested methodologies for iterative creation and analysis of early RE agent-goal models. The methodology specifically guides users on how to start analysis for newly created models, including the types of default "sanity" check questions that can be asked over models.

Implementation. Framework components are implemented in the OpenOME requirements modeling tool. Our OpenOME implementation supports the forward and backward analysis procedure, including storage of multiple evaluation results, management of human judgments, and analysis visualizations

Framework Validation. The forward procedure component of the framework has been tested via several case studies, including a demonstration of the differences between proponents and opponents of Trusted Computing Technology (Horkoff, Yu, & Liu, 2006), as part of a method to analyze the effectiveness of knowledge transfer, and an analysis of online counseling in a large social service organization, with selected results reported in (Easterbrook et al., 2005; Strohmaier, Horkoff, E. Yu, Aranda, & Easterbrook, 2008). A small experiment was conducted over models describing conference "greening" in order to find evidence to support the perceived contributions of the analysis procedure (Horkoff & E. Yu, 2009a; 2010b).

In addition, we have administered two types of case studies to test the contributions of interactive analysis (Horkoff & E. Yu, 2010d). We have conducted ten individual case studies using subjects with some experience in i* modeling. Half of the participants analyzed models using no explicit procedure (ad-hoc analysis) while the other half used implementations of the forward and backward interactive analysis procedures. Five-follow-up studies were conducted to test analysis visualizations. In order to gain some insight into analysis by individuals versus analysis in a group, we administered a separate multi-session case study involving a project team designing tool support for modeling "back of the envelope" calculations. Qualitative and quantitative analysis of results are used to compare treatments in both studies, to gather evidence to support or deny claims, and to gain an understanding of the benefits of and barriers to systematic goal model analysis.

1.5 Framework Contributions

We outline the contributions provided by the framework developed in this work, using the challenges in listed in Section 1.2 to frame contribution descriptions.

Model Complexity. The framework in this work balances between the need for automation due to model complexity with the need for interaction and human intervention to account for the incompleteness and relative accuracy of complex early RE models. Visual interventions help modelers to understand analysis results over complex models. Individual studies have collected evidence to support the utility of these visualizations.

Model Completeness. The interactive nature of the analysis framework forces modelers to examine key pieces of the model when making judgments over conflicting or partial evidence. We argue that this process can lead to the discovery of important model omissions, and can urge iterative improvement of model completeness. Suggested modeling methodologies instruct users on how to use the analysis procedures to test the basic "sanity" of the models, including completeness. We develop and administer various studies to test these claims.

Model Accuracy. As modelers are making judgments over fragments of the model, they are inherently checking the accuracy of model contents. When the interactive component of the framework is performed in a group session, judgments over fragments of the models can result in consensus building and subsequent modification of the model to represent this consensus. Model analysis results can also be used to check the accuracy of the model. If results are surprising, then either new a discovery concerning the domain have been made, or the contents of the model are not sufficiently complete or accurate.

Domain Understanding. We claim that the interactive nature of the framework introduced in this work can encourage further elicitation in the domain, increasing domain knowledge and improving model accuracy and completeness. Experiential and concrete evidence to support these claims, including the perceived conditions under which this elicitation may occur, are explored as part of the framework validation.

Model Interpretation. By formally defining propagation rules over i*, we have enforced a more consistent interpretation of this particular syntax. Results of the validation studies show that systematic analysis produces a more consistent model interpretation when compared to ad-hoc (without a systematic procedure) analysis. Increasing the consistency of model interpretation will help to build a more consistent consensus amongst stakeholders involved in early RE elicitation.

Model Flexibility. By expanding on methods which allow for qualitative analysis over high-level domain concepts, the framework provides flexibility in modeling and analysis. The analysis procedures work over flexible, non-quantitative goal model frameworks (i.e. i*), and do not require the addition of extra, more detailed information (e.g., costs, temporal ordering) to produce analysis results.

Capturing Rationale for Decisions. By capturing and managing user judgments over contentious or non-functional areas of the model, we have provided support for understanding the rationale behind key early RE decisions. Further improvements could allow users to enter additional text to describe the reasoning and assumptions behind their decisions.

The Role of the Stakeholder. The incomplete and imprecise nature of early RE models means that active participation of stakeholders in a cycle of model improvement and elicitation is key for an accurate and effective high-level view of the system-to-be. The analysis procedures in the framework encourage interactive analysis by acquiring human judgment to resolve partial or conflicting evidence. This increases the role of the stakeholder in the analysis process, potentially giving them a sense of ownership in the model, analysis results and subsequent decisions made.

Analysis Power. The framework defined in this work has provided two unique forms of analysis, forward analysis allowing users to as "what if?" questions, and backward analysis allowing users to ask "Is this possible?", "If so, how?" and "If not, why?". Detailed comparisons of forward analysis procedures have made it clear that analysis results over E\early RE goal models should be treated as heuristics, and not taken at face value without intensive stakeholder involvement. Methodologies provided with the framework guide users in how to use qualitative analysis to iteratively and interactively understand the domain through models.

Procedure Selection. Our survey of goal model analysis procedures has included a consideration of analytical capabilities and information required to perform analysis. The detailed comparison of forward satisfaction procedures has provided a better understanding of the differences between existing approaches in this area, including criteria which could be used to select between methods. Using this information, a potential user of goal models could use domain-related knowledge to decide whether existing procedures would fit their purpose, or whether the framework more specifically intended for early RE analysis described in this work would be more appropriate.

Generalizability. The framework developed in this work uses the i* Framework as an example agent-goal model framework. This framework was selected as it was specifically intended for early RE analysis, and the modeling framework is generally is a syntactic superset of several related frameworks (e.g., NFR, GRL and Tropos). Although we focus on application to early RE goal model analysis, the general theme of iteration and interaction could be useful for other types of models used for other purposes. Some components of the framework could be reused and adopted in other contexts.

1.6 Thesis Organization

The remainder of this document is organized as follows. Chapter 2 reviews existing approaches for analysis of goal models, including a summary of related work in RE and business, then describes criteria for selecting existing goal model analysis procedures, depending on the problem being addressed and the information available. Chapter 3 outlines the requirements for an effective agent-goal modeling analysis framework for early RE, then evaluates existing goal model analysis procedures in light of these requirements. Chapter 4 provides a detailed comparison of forward satisfaction analysis techniques for goal models, using specific examples from the literature. Chapter 5 reflects on the syntax of our modeling framework of choice, i*, providing a formal definition of model syntax. Chapter 6 begins to describe the framework by providing the details of the forward analysis procedure. Chapter 7 provides details on the backward analysis procedure. Chapter 8 describes a suggested methodology for model creation and both forward and backward analysis. Chapter 9 describes visualizations provided to help the modeler apply analysis and understand analysis results. Methods to find inconsistencies amongst user judgments and between judgments and the model are described in Chapter 10. Chapter 11 describes the tool support implemented to support the framework, including procedure performance. Chapter 12 describes validation of the framework through multiple case studies and experiments. Chapter 13 summarizes the contributions of the framework in light of the requirements outlined in Chapter 3, describes framework limitations, and outlines future work.

Chapter 2 Literature Survey

In order to set the groundwork for the framework for early RE agent-goal model analysis introduced in this work, we review and assess existing goal-oriented modeling and analysis techniques. A great variety of techniques for analyzing goal models in a requirements engineering context have been proposed. Approaches include propagating goal satisfaction values, computing metrics over models, finding acceptable models using planning algorithms, simulating model behavior, and checking formal properties over a model. In this Chapter, we survey available approaches for goal model analysis. As part of our survey, we aim to answer the following questions:

1. Survey of methods: What goal model analysis methods are available? What types of analysis questions can these methods answer?
2. Modeling constructs: What goal model constructs or notations do the procedures support? Information: What domain information is needed in order to use the methods?

Several approaches not specific to goal models can also provide insights to Early RE analysis. Specifically, this chapter briefly covers some approaches in the fields of Requirements Engineering and Business. In addition, we consider alternative means of supporting and reasoning over design decisions, such as qualitative reasoning, and multi-valued logic approaches.

Goal model analysis techniques may be used to meet a variety of objectives encountered throughout the software lifecycle. These objectives may be applicable to early RE stages, or may apply, for example, to later RE, design, implementation or maintenance. However, from a practical viewpoint, the diversity in goal model analysis techniques creates a barrier for widespread adoption of such techniques.

This chapter is an expansion of the following papers/reports:

Horkoff, J. (2008a). Analysis of Goal- and Agent-Oriented Models, Depth Oral Report.

Horkoff, J., & Yu, E. (2011a). Analyzing goal models: different approaches and how to choose among them. Proceedings of the 2011 ACM Symposium on Applied Computing (pp. 675-682). New York, NY, USA: ACM.

Recognizing the lack of guidance to the literature and how to choose among these techniques, this chapter offers a first attempt to suggest initial guidelines for choosing techniques to meet analysis objectives. Specifically, we aim to answer the following questions:

3. Analysis Benefits: What are some of the potential benefits of goal model analysis in the requirements process?
4. Fitness for purpose: Which available methods can be applied to achieve which kinds of usage objectives?
5. Selection: How can we use this information to advice on general selection?

In the second part of this chapter, we look broadly at the objectives which may be met by goal model analysis techniques throughout the software lifecycle. We map these objectives to reviewed procedures, and then use this mapping to provide guidelines for procedures selection, based on domain characteristics and lifecycle stage. We then provide examples which apply these guidelines to select procedures for two different case studies.

2.1 Goal Modeling Frameworks

Goal-Oriented Requirements Engineering (GORE) frameworks allow for the representation of one or more goals, which may be derived from the system or system stakeholders, and which may have relationships to other goals, often describing how a goal can be achieved, or if a goal negatively impacts other goals. Example goal modeling frameworks, techniques or methodologies include KAOS, GBRAM, NFR, i*, Tropos, GRL, and AGORA described briefly below.

The KAOS Methodology introduced a formal goal framework applying AND and OR decompositions between goals describing desired states over entities, achieved by actions assigned to agents (Dardenne, van Lamsweerde, & Fickas, 1993). The GBRAM technique guides the elicitation of goals from system activities and artifacts, classifying goals, and associating them with constraints and scenarios (Anton & Potts, 1998). Goals in GBRAM are refined using questions and scenarios, and are represented in tabular form.

The NFR (Non-Functional Requirement) modeling framework aims to represent human intentions in technical systems (Chung et al., 2000). The framework uses the concept of *softgoals,* goals that are not satisfied via clear-cut criteria, AND and OR decompositions amongst goals, and *contribution links*, representing potentially partial negative and positive contributions to and from such goals. The i* (distributed intentionality) framework incorporates concepts from the NFR Framework, including softgoals, AND/OR decompositions, and contribution links (Yu, 1997). In addition to NFR syntax, i* contains (hard) goals, resources, and dependencies between actors (agents). The i* framework is used as a first stage in Tropos, an agent-oriented system development methodology (Bresciani, Perini, Giorgini, Giunchiglia, & Mylopoulos, 2004). A simplified version of i* was used to create GRL (Goal-oriented Language), which together with Use Case Maps (UCM) constitute URN (User Requirements Notation), recently approved as an ITU-T international standard (Amyot, 2003; ITU-T, 2008; Liu & E. Yu, 2004).

The Annotated Goal-Oriented Requirements Analysis (AGORA) approach addresses missing capabilities of existing goal-oriented approaches by including goal priorities and methods for solving goal conflicts, selecting alternatives, and measuring the quality of models (Kaiya et al., 2002).

The Business Intelligence Model (BIM) adopts ideas introduced by existing goal model frameworks, linking them to real data sources for the purpose of business intelligence (Barone et al., 2010). Current business intelligence systems use real data fed into indicators as a means to monitor the current status of a business. Indicators are often displayed on a "dashboard", showing whether key points of measurement are currently over or near certain thresholds (often represented using colors like red, orange and green). The BIM model aims to bridge the gap between these systems and work in goal and process models, adding a level of intentionality to data analysis.

2.2 Survey Design

In this section, we describe methods for selecting papers as part our survey and motivate use of the classification schemes used to categorize existing work.

2.2.1 Paper Selection: Criteria and Methods

In this survey, we focus on systematic analysis procedures over primarily graphical goal model representations consisting of goals and relationships. We limit our survey to analysis procedures which work over models which minimally support a set of goals linked together by AND/OR and some kind of contribution links. We chose to focus on this type of goal model as it allows analysis of properties using the relationship between goals. We focus on analysis procedures which use the structure and the relationships of the model to derive useful information such as the effects of alternative designs or the satisfaction level of critical domain properties such as security.

Articles in this survey were collected by means of linking work through references. An initial seed set of articles known to be related to goal model evaluation was collected, relevant work referenced by these articles were examined for relevance. The cycle continued until a picture of the breadth of goal analysis methods was obtained. These works cover conferences/journals/workshops in several areas (e.g., Requirements Engineering, Software Engineering, Agent Systems, AI, Enterprise Modeling, Information Systems, Trust, and Security) and employ a host of different keywords (e.g., agent-oriented software development, goal-oriented requirements analysis, early requirements analysis, multi-agent systems, agent-oriented software engineering, agent-oriented methodologies, risk analysis, countermeasure identification, goal modeling, requirements elicitation, goal oriented analysis, and quality metrics). Our findings have indicated that an alternative method of systematic article selection (i.e. by specific journals and/or keywords) would not be as successful in finding relevant articles. The survey is not intended to be complete, but offers a useful overview of prominent goal model analysis work.

2.2.2 Technique Classification

Goal model analysis approaches are presented in categories according to the techniques used (Satisfaction Analysis, Metrics, Planning, Simulation and Model Checking). This categorization is used as it is closely related to the type of analysis questions facilitated by the procedures. However, this division is not clear-cut, as many techniques employ more than one approach. Survey results are further summarized over several points using tables at the end of each section. The algorithm approach taken by each of the works is summarized in the first columns of each table. The satisfaction analysis category is divided into forward and backward propagation directions. For each work/algorithm combination, we have entered Y (Yes, uses this approach), N (does not use this approach) or M (Maybe, not clear whether it uses this approach or not). An extra category has been added to capture the need for human intervention -- whether or not the procedure is interactive and requires expert or stakeholder intervention to produce analysis results.

We note that several of the procedures make different choices over the form of measurement for analysis results. Some procedures produce qualitative results, others quantitative, others binary (yes/no answers), while some procedures can produce different results in more than one of these forms. The selection of measurement scale is significant as it shapes the type of answers each procedure can provided to analysis questions. Binary measurements can provide only yes/no answers, qualitative procedures can provide an ordinal scale of property satisfaction, while quantitative procedure can provide more precise measures. However, the accuracy of quantitative measures depends on the accuracy of the input measures, models, and calculation method. We summarize the type of analysis result in the Analysis Results columns of each table.

We have defined goal models of interest to our survey as containing a minimum of goals and AND/OR decompositions. Many of the procedures we have reviewed support analysis over additional goal model syntax. Some of the most commonly supported syntax includes softgoals, contribution links, actors and dependencies between actors. The type of syntax supported is significant in that is affects the types of analysis questions which can be answered using the model. Support for such syntax is summarized in the last columns of each summary table. The final table incorporating all classifications is shown in Table 38.

Our summaries also make use of the distinction between global and local analysis. This distinction is often tied to the use of agents, with global analysis being a consideration of goal satisfaction across the entire model, and local analysis considering the satisfaction of a particular agent, or a particular sub-section of a model. Although global analysis can give you a high-level view of all the effects in a model, local analysis can provide an actor-centric, simpler view, better dealing with model incompleteness.

We also identify information beyond these notational constructs which is required by various procedures to perform analysis. For example several procedures ask modelers to enter information regarding the cost of goals, while other procedures want information concerning the relative priorities of each goal. The distinction between additional syntax and such additional information can be blurry, for example in the goal models used for the simulations in (Gans et al., 2004) pre- and post- conditions for goals are drawn graphically on the model using triangular shapes. Visual inclusion of such additional information often differs between techniques. In contrast, the items we identify as syntax (softgoals, contribution links, and dependencies) are used in several different analysis techniques and often have a common graphical representation. Categorizing procedures by additional required information can aid in selection; if information is not readily available, a procedure cannot be easily used. We explore the additional information required for each procedure during our review, and summarize this information in Table 39 in Section 2.3.7.

2.3 Goal Model Analysis Techniques

Several approaches introduced for the analysis of goal- and agent-oriented models are aimed to measure the satisfaction or achievement of goals, while others are intended to measure other specific properties such as predictability or risk. Tools and methodologies have been developed which to apply planning or simulation to agent-oriented goal models, attempting to find effective system configurations or to detect problems in the high-level system design. We review and classify each type of method in the following sections.

2.3.1 Satisfaction Analysis

This section focuses on methods whose result is given in terms of a measure of the satisfaction (and denial) of goals in the model. Such measurements are derived given the satisfaction or denial of other elements in the model, often leaf elements. The section is further divided into procedure which support binary, qualitative and/or quantitative analysis.

2.3.1.1 Binary Satisfaction Analysis

Several approaches find solutions in a goal model using formal methods in which goals are either satisfied or not (binary satisfaction). We summarize several of these approaches in this section.

(Maiden, Lockerbie, Randall, Jones, & Bush, 2007): Maiden et al. argue that i* modeling contains two major faults, that means-ends link (including contribution links) do not help to capture the underlying reasons behind these connections, and that i* does not link well to existing Requirements

Engineering processes. To address the problems, the authors use the idea of a satisfaction argument, introduced in (Zave & Jackson, 1997), where using relevant domain properties (D) along with a specification of system behavior (S), can show (|-) that requirements (R) will hold.

They map these concepts to i* model concepts, with R mapped to ends in means-ends and contributions, and S mapped to means. Structured text is used to justify the satisfaction of Requirements in the model. Existing requirements from a specification are mapped to tasks, determining whether existing requirements satisfy tasks using a matrix showing the contribution of requirements to tasks. Task and resources are judged to be compliant or potentially non-compliant, based on the results of the matrix, and these effects are propagated through the models using rules, called heuristics here, which have similarities to the rules described in (Horkoff, 2006). After the propagation, potentially non-compliant goals are made compliant or non-compliant by examining and rewriting the satisfaction arguments. This method considers whether or not elements in the model are compliant, without considering varying degrees of compliance or satisfaction. They apply their ideas to a case study involving a tool to warn about air traffic infringements.

Classification: We classify this approach in Table 1 as using forward satisfaction analysis. It is classified as binary under Measurement Treatment as it does not consider degrees of satisfaction, using either a quantitative or qualitative scale. This method uses human intervention to determine compliance and evaluates models using an agent-oriented paradigm, propagating globally across models. The method uses supplementary information in the form of textual arguments, or satisfaction arguments, to justify compliance, as included in Table 38.

Table 1 Classification of (Maiden et al., 2007)

	Approach							**Analysis Results**			**Additional Notation Supported**			**Analysis Scope**	
Approach	**Satisf Forwds**	**Satisf Backwds**	**Human Interv**	**Metrics**	**Plan-ning**	**Simu-lation**	**Model Check**	**Qual**	**Quant**	**Binary**	**Depend-encies**	**Soft-goals**	**Contrib. Links**	**Global**	**Local**
Maiden et al., 2007	Y	N	Y	N	N	N	N	N	N	Y	Y	Y	Y	Y	N

(Jureta, Mylopoulos, & Faulkner, 2008): In this work, the authors argue that the S, K |- R ontology (Specification (S) and Domain Knowledge (K) must be sufficient to guarantee that Requirements (R) are satisfied) introduced by Zave & Jackson (1997) is incomplete, missing a consideration of beliefs, desires, intentions, and attitudes. They argue that the concepts do not allow for a consideration of the partial satisfaction of requirements or a comparison of alternatives in terms of preferences. They expand this ontology formally by considering speech act theory, and mapping classifications of speech acts uttered by the stakeholders to concept in their ontology. Namely, assertive or declarative speech acts are mapped to beliefs, directive speech acts are mapped to desires, commissive speech acts are mapped to intentions, and expressive speech acts are mapped to attitudes. Desires are further divided into quality constraint goals and softgoals, with quality constraints being precisely measureable and softgoals not. The notion of justified approximation is introduced to find quality constraints that justifiably approximate softgoals. In other words, softgoals eventually become precisely measurable.

Intended content is grouped together as plans. All plans that satisfy the goals and quality constraints of the system, given the domain assumptions, can be compared using stakeholder's attitudes. They define attitudes over all concepts in their ontology and distinguish between optional and preference roles for attitudes. Plans must include all non-optional entities, and the plan which best satisfies the preferences are selected. In this way the method helps to select between design alternatives.

Although this method makes a valid point about the exclusion of some concepts in the original formalization and S, K |- R ontology, the ontology it introduces is complex and may be difficult to apply. Specifically, classifying all stakeholder communications as speech acts would be quite laborious and tedious. Related work uses satisfaction arguments to justify the satisfaction of requirements given D and S in an i* model (Maiden, Lockerbie, Randall, Jones, & Bush, 2007). Such arguments take into account NFRs in a more informal, simple, and argumentative way. Such an approach may be more appropriate for very early analysis.

We classify this work along with the work below.

(Jureta, Borgida, Ernst, & Mylopoulos, 2010): Jureta et al. introduce the Techne formal requirements language in order to provide a consistent ontological foundation and basic reasoning for all early Requirements Modeling Languages (RML). They create a language which covers concepts in their core ontology for early RE (goals, softgoals, quality constraints, assumptions, tasks, preferences and optional requirements). The language uses the r-net visual syntax, but allows new RMLs to adopt the formal underpinning provided by the language, adding their own visual interpretation. In addition to the entities supported by the language (goals, softgoals, quality constraints, assumptions, and tasks), the language defines several relationships: inference (implication), conflict, preference, is-mandatory, and is-optional. The term "requirement" is used here to represent any of the concepts in the ontology.

As in Jureta et al. (2008) softgoals must be approximated. Here, they are approximated by an approximation which is a set of requirements. Differences between the level of softgoal satisfaction by various alternatives are captured indirectly by preferences over different approximations.

The language supports analysis allowing users to find the candidate solutions to a requirements problem as defined in Techne, and the preferred and optional requirements in each solution. For the first question, the approach attempts to find maximal consistent subsets of the problem such that tasks are source nodes (leaves) and mandatory requirements are satisfied. For each candidate solution, the optional and preferred requirements can be compared, for e.g. using a table. How to select between these solutions is not described in this work.

This work provides a conceptually useful way to frame the requirements problem. It deliberately avoids collecting quantitative measures of prioritization in order to be more amenable to early RE. Generally, the inclusion of preferences, optional and mandatory goals can be useful, but it is difficult to know how easy it is to gather this information from the stakeholders early in the analysis process. Pairwise comparison of requirements needed to gather preferences may not be realistic in large, early models. The insistence that all softgoals must be approximated makes the approach less applicable for very early RE. Although degree of goal satisfaction can be expressed indirectly via preferences, it is arguable that this approach may not scale. For example, if a user prefers goal *a* over goal *b* because *a* is better for softgoal *c*, what if *b* is better than *a* for a softgoal *d*? Which is more preferred?

The approach does not focus on the construction of the model, how to aim for model completeness and accuracy. Instead it focuses on how to describe and reason over concepts, once they have been gathered. It is not clear if applying this form of analysis would lead users to recognize when their model is incomplete or inaccurate. We believe this type of analysis is better suited to a later stage of requirements analysis, when confidence in the contents of the model is higher.

Classification: This work uses binary analysis which finds candidate solutions in which all mandatory requirements are satisfied. The analysis approach is classified as backward, although it is not explicitly called so in the work, as it finds all solutions using the structure of the model, instead of looking for the effects of choosing specific alternatives ("what if?"). The approach supports use of softgoals, but not contribution links, as all softgoals are approximated, allowing for binary analysis.

Table 2 Classifications for (Jureta et al., 2008) and (Jureta et al., 2010)

	Approach							Analysis Results			Additional Notation Supported			Analysis Scope	
Approach	**Satisf Forwds**	**Satisf Backwds**	**Human Interv**	**Metrics**	**Plan-ning**	**Simu-lation**	**Model Check**	**Qual**	**Quant**	**Binary**	**Depend-encies**	**Soft-goals**	**Contrib. Links**	**Global**	**Local**
Jureta et al., 2008, 2010	N	Y	N	N	N	N	N	N	N	Y	N	Y	N	Y	N

Discussion: Binary satisfaction provides a simple, clear-cut way to view the satisfaction or denial of goal in the domain, forcing users and procedures to make a definite decision over goal satisfaction. However, these procedures lack expressiveness for the notion of partial satisfaction or denial, capturing the presence of partial positive or negative evidence for the satisfaction of a goal. These notions are especially useful in early RE analysis which often involves high-level objectives which are not clear-cut. A combined summary of the binary analysis procedures can be found in Table 3.

Table 3 Summary of Classifications for Binary, Forward Analysis Procedures

	Approach							Analysis Results			Additional Notation Supported			Analysis Scope	
Approach	**Satisf Forwds**	**Satisf Backwds**	**Human Interv**	**Metrics**	**Plan-ning**	**Simu-lation**	**Model Check**	**Qual**	**Quant**	**Binary**	**Depend-encies**	**Soft-goals**	**Contrib. Links**	**Global**	**Local**
Maiden et al., 2007	Y	N	Y	N	N	N	N	N	N	Y	Y	Y	Y	Y	N
Jureta et al., 2008, 2010	N	Y	N	N	N	N	N	N	N	Y	N	Y	N	Y	N

2.3.1.2 Qualitative Satisfaction Analysis

Several approaches which aim to measure the satisfaction and denial of goals, given a particular situation or design option, do so using a qualitative approach. We distinguish between qualitative and quantitative approaches in each summary table under the analysis results columns.

(Chung, Nixon, Yu, & Mylopoulos, 2000): A qualitative evaluation procedure was introduced as part of the NFR Framework with the high-level intention of allowing evaluation of design alternatives in respect to the non-functional requirements of the system, helping to choose the alternative which involves the best tradeoffs between system goals. To this end, labels are placed on the graph to indicate the selection of an alternative, these labels are propagated throughout the graph, and the results are analyzed.

The procedure uses the concepts of "satisficed" and "denied" to represent sufficient evidence of goal satisfaction, and sufficient evidence of goal denial, respectively. These labels are also used to initiate the procedure by selecting design alternatives. The procedure uses six qualitative labels to represent fully satisficed (✓), weakly satisfied (W^+), undetermined (U), conflict (♮), weakly denied (W^-) and denied (X). Weak satisfied/denied refers to the situation where there exists positive/negative evidence towards the satisfaction/denial of a goal, but this evidence is not sufficient enough to judge the goal as fully satisficed/denied. Undetermined represents the case where no evidence is available. Conflict indicates that an element is both satisficeable and deniable.

In the procedure, the initial labels are propagated from offspring to parent goals using both propagation rules and human judgment. These rules indicate what labels are propagated, given the label of the offspring and the type of link. AND contribution links propagate the minimum value, using the ordering:

♮ ≤ U ≤ X ≈ ✓,

while OR links propagate the maximum value. Propagation through other contribution types (+, ++, -, -) is described in Table 4, recreated from Table 3.2 in (Chung et al., 2000).

Table 4 The "individual impact" of an offspring on its parent during the First Step of NFR evaluation, recreated from Table 3.2 in (Chung et al., 2000)

Individual Impact of offspring with label:	**Upon parent label, given offspring-parent contribution type:**							
	Break	**Some-**	**Hurt**	**?**	**Help**	**Some+**	**Make**	**=**
X	W^+	W^+	W^+	U	W^-	W^-	X	X
♮	♮	♮	♮	U	♮	♮	♮	♮
U	U	U	U	U	U	U	U	U
✓	X	W^-	W^-	U	W^+	W^+	✓	✓

The procedure consists of two steps. In step one all current values are propagated from offspring to parent using the propagation rules. Goals which receive an unknown or conflict label require human intervention. As a softgoal may receive more than one label via more than one contribution link, these labels must be combined into a single label, possibly requiring human judgment. Step two involves the resolution of these softgoal labels. The procedure suggests collecting the labels for one parent node in a bag, allowing for duplicates. In some cases, when the result is clearly satisficed, denied, or conflict, the incoming labels can be combined automatically. In other cases, human intervention is required. The work recommends that all partial values are combined together into one or more full, unknown or conflict labels, and that the final result is the minimum of these combined labels, using the ordering above. If both a satisficed and denied label remain, the result is a conflict. These steps are then repeated until all values have been propagated.

The human judgment required in this procedure is a point of interest. It is up to the analyst to promote or demote values, and to try to resolve conflicting values. This process should make use of domain knowledge, including knowledge of the relative importance of each offspring.

Although the rules given describe the promotion or demotion of all partial values, it is mentioned that the procedure can be expanded to allow for partial values as a final label of a node. The last part of the description outlines how the procedure could be modified to allow for weak labels as results, but does not specify the full details of this adjustment.

Lamsweerde, (2009), summarizes the qualitative propagation introduced as part of the NFR Framework. Although the work pays tribute to the use of softgoals as a means to compare alternative options, it points out several flaws with the qualitative reasoning procedure. Specifically, resulting labels often become rapidly inconclusive, resulting often in unknown labels; labels and link weights have no real meaning in the domain, producing inaccurate results; and, all leaf soft goals are assumed to have the same priority in the domain. To address these problems van Lamsweerde introduced a quantitative reasoning procedure using gauges over the model. We summarize this procedure in Section 2.3.1.3.

Classification: This procedure is categorized as providing forward satisfaction analysis using human intervention with qualitative and binary measurement results. We use the binary category to describe evaluation in terms of AND and OR decomposition, where a goal can only be satisfied or not. The procedure works over models with softgoals and contribution links, but not with dependencies. It evaluates on a global scale, across an entire connected model.

Table 5 Classifications for (Chung et al., 2000)

	Approach							Analysis Results			Additional Notation Supported			Analysis Scope	
Approach	**Satisf Forwds**	**Satisf Backwds**	**Human Interv**	**Metrics**	**Plan-ning**	**Simu-lation**	**Model Check**	**Qual**	**Quant**	**Binary**	**Depend-encies**	**Soft-goals**	**Contrib. Links**	**Global**	**Local**
Chung et al., 2000	Y	N	Y	N	N	N	N	Y	N	Y	N	Y	Y	Y	N

(Giorgini, Mylopoulos, Nicchiarelli, & Sebastiani, 2002; 2004a): Giorgini et al. have introduced a qualitative evaluation procedure which contains similarities to evaluation with NFR models. Both the NFR and the Giorgini et al. procedures are qualitative and propagate evidence "bottom-up", propagating labels from leaf goals to higher-level goals. However, differences between the representation of satisfaction and denial, the syntax of the target models, and the algorithm exist.

In terms of syntactical differences, goal models targeted by the Giorgini et al. procedure contain events, observable goals which feed values into the goal graph. In these models, there is no explicit use of the idea of a softgoal; all goals can take on partial evaluation values. These models allow for non-symmetric contribution labels. Labels of S or D on a link indicate that only positive or negative evidence is propagated, respectively. An absence of any letter on the link indicates that the values are propagated symmetrically, meaning both positive and negative evidence is propagated.

In the Giorgini et al. procedure, the degree of satisfaction or denial is represented as a predicate over a goal. These predicates range from full evidence of satisfaction, FS, to partial evidence of satisfaction, PS, to partial evidence of denial, PD, to full evidence of denial, FD. In this procedure, the term "satisfaction" is used to mean that there is at least full evidence that a goal is satisfied. Each goal is assigned two variables, Sat and Den, over the range of {F, P, N}, representing the level of evidence for the satisfaction and denial of a goal, with F, P, and N representing full, partial, and none, respectively. The predicates FS(G), PS(G), PD(G), and FD(G) are defined as Sat(G) >= F, Sat(G) >= P, Den(G) >= P, and Den(G) >= F, respectively.

The separate formalization of positive and negative evidence allows the procedure to be fully automated by a set of propagation axioms which define how predicate values are propagated through links. Conflicts, the presence of both negative and positive evidence, are propagated and not resolved. Human intervention is not used to resolve evaluation values. The propagation rules implemented by the axioms are the same as the rules described in (Chung et al., 2000), modified to account for the separation of positive and negative evidence.

The algorithm introduced in this procedure is guaranteed to converge after 6*|G|+1 iterations, where |G| is the size of the graph. This is possible as the algorithm propagates non-decreasing labels. The algorithm is applied to artificially-generated large examples, and is shown to converge in a reasonable amount of time.

In this approach, the presence of both negative and positive evidence automatically indicates a conflict, and such conflicts are not resolved. When faced with multiple partial contributions of the same polarity to a single goal, there is no mechanism to promote these values to a full value. As a result of these conventions, it is common to find results where nearly all higher-level goals have conflicting values, a result which is difficult to interpret.

Classification: In our Table 6 classification, the Giorgini et al. procedure is classified as a qualitative algorithm measuring satisfaction. The algorithm works in a forwards direction and does not involve human intervention. The procedure works over models with contribution links, and although softgoals are not explicitly mentioned, a goal with an incoming contribution link can be considered implicitly "soft". To reflect this we add an "M" (maybe) in the Softgoals column under Additional Notation Supported. The procedure evaluates globally across and entire connected model. A quantitative version of this approach is available, as described in Section 2.3.1.3.

Table 6 Classification of (Giorgini et al., 2002; 2004a)

	Approach							Analysis Results			Additional Notation Supported			Analysis Scope	
Approach	**Satisf Forwds**	**Satisf Backwds**	**Human Interv**	**Metrics**	**Plan-ning**	**Simu-lation**	**Model Check**	**Qual**	**Quant**	**Binary**	**Depend-encies**	**Soft-goals**	**Contrib. Links**	**Global**	**Local**
Giorgini et al., 2002, 2004a	Y	N	N	N	N	N	N	Y	Y	Y	N	M	Y	Y	N

(Giorgini, Mylopoulos, & Sebastiani, 2004b): In further work, this approach is expanded to produce a top-down analysis using a SAT solver (Giorgini et al., 2004b). This procedure searches for combinations of lower-level input goals which would produce the desired high-level values (targets). The scheme of predicates over goals with Sat and Den variables, as well as the propagation rules, remains the same as in previous work. However, the propagation in this method is intrinsically more complex, as the search for lower-level values producing acceptable higher-level values can produce multiple possibilities.

The procedure uses propositional satisfiability (SAT), the problem of finding a satisfying assignment of variables in a Boolean formula. It uses a state-of-the-art SAT solver which makes calculation practical, despite the NP-complete nature of the problem. The structure of the goal graph, the desired set of goal values, as well as the axioms for backwards propagation are converted into a formula, which can be used as input for the SAT solver. In addition, this method allows for the addition of constraints and a restriction of conflict levels.

In order to deal with graph cycles, the approach has a limitation that every target goal of the procedure must have a direct acyclic sub-graph whose leaves are input goals. Furthermore, all leaf goals of the graph, goals with no incoming links, must be input goals for the algorithm.

Constraints on values can be added by specifying whether or not certain goals should take on certain values, for example: PS(G1) or not FD(G2). Conflict avoidance can be specified by avoiding all strong conflicts (FS and FD), all strong and medium conflicts (FS and PD or FD and PS) or all conflicts (PS and PD). This procedure also allows for a consideration of costs. Leaf goals in the goal graph can be assigned relative costs. The evaluation procedure is adjusted to find the minimum cost solution which will satisfy constraints and result in the desired top-level values.

The termination of this procedure relies on the termination of the corresponding SAT problem. However, the implementation was tested on artificially created goal graphs and was shown to produce results in a reasonable amount of time for graphs of size less than three-hundred.

Classification: We add a classification for the (Giorgini et al. 2004b) procedure in Table 7. The categorization is the same as that of (Giorgini et al., 2002; 2004a) except the procedure works in a backward direction in the Approach column and the removal of quantitative as a Measurement Treatment. In Table 38, we record the incorporation of cost measures into the model.

Table 7 Classification of (Giorgini et al., 2004b)

	Approach							Analysis Results			Additional Notation Supported			Analysis Scope	
Approach	**Satisf Forwds**	**Satisf Backwds**	**Human Interv**	**Metrics**	**Plan-ning**	**Simu-lation**	**Model Check**	**Qual**	**Quant**	**Binary**	**Depend-encies**	**Soft-goals**	**Contrib. Links**	**Global**	**Local**
Giorgini et al., 2004b	Y	Y	N	N	N	N	N	Y	N	Y	N	M	Y	Y	N

(Giorgini, Mylopoulos, & Sebastiani, 2005): Work in (Giorgini et al., 2005) expands on previous work by incorporating goal model analysis procedures into the Tropos Framework. Forward reasoning is used to evaluate alternatives in the system to-be model used in the late RE stage of Tropos. Backwards reasoning is used to find the acceptable alternative with the lowest cost.

This work introduces the concept of loops, a path from a goal to itself. Using this concept, they relax the restriction of goal model construction introduced in (Giorgini et al., 2004b) by saying that every loop must have at least one arc which is a contribution link.

Adaptation of goal model analysis techniques for Tropos, which includes syntax for actors, requires a consideration of the agent-oriented paradigm. Here the analysis described in the authors' previous work is confined to intra actor analysis, involving only the goals of a particular actor. Inter actor analysis, analysis across actor boundaries, is mentioned only briefly, and performed by extending the boundaries of the formalism outside of the considered actor. The specifics of how to extend the formalism to cover dependency relationships are not provided; however, these relationships could possibly be considered as ++ (make) relationships. Further investigation in needed to determine if this extension would be sufficient.

The paper describes the GR-Tool, implementing the forward and backward algorithms in more detail. Backward reasoning is aided by the GOALMINSOLVE Tool.

Classification: This work is classified similarly to previous work by Giorgini et al. This particular paper focuses on the qualitative procedure, and uses both forward and backward reasoning. Approaches reviewed up to this point have considered goal decompositions without considering dependencies between agents. Although the agent-oriented paradigm is considered, details on how to expand the propagation across dependencies between actors are not given. We add an "M" to the Dependency column for this procedure. Analysis in this procedure focuses on the per-actor view, with global analysis described only briefly. In all other works considered so far, only a global viewpoint of satisfaction in considered.

Table 8 Classification for (Giorgini et al., 2005)

	Approach							Analysis Results			Additional Notation Supported			Analysis Scope	
Approach	**Satisf Forwds**	**Satisf Backwds**	**Human Interv**	**Metrics**	**Plan-ning**	**Simu-lation**	**Model Check**	**Qual**	**Quant**	**Binary**	**Depend-encies**	**Soft-goals**	**Contrib. Links**	**Global**	**Local**
Giorgini et al., 2005	Y	Y	N	N	N	N	N	Y	N	Y	M	Y	Y	M	Y

(Ernst, Mylopoulos, Borgida, & Jureta, 2010): Ernst et al. extend the approach Jureta et al. (2010), describing how to select between candidate solutions using mandatory/optional requirements and preferences over requirements. They utilize and expand the qualitative reasoning described by (Giorgini et al., 2004b) in order to include the option of mandatory (must be fully satisfied FS(goal)), optional requirements, and preferences.

The approach allows users to find candidate solutions which would satisfy all of the mandatory requirements, as in Jureta et al. (2010). Then, the procedure attempts to find the maximum number of optimal goals which can be added to the solution without breaking the satisfaction of mandatory goals. Optional goals are pruned using conditional knowledge of applicability and preferences, removing options which are dominated by other options. Efficient algorithms for selecting optional requirements are explored, including a local search. Finally, the set of possible solutions including optional goals are pruned using preferences. The result is one or more dominant solutions.

This approach provides a useful contribution by allowing users to select between alternatives solutions using optionality and preference; however, the approach assumes that the models under analysis are sufficiently complete and accurate and does not describe how analysis results could lead to model iteration or improvement. Although this work supports qualitative analysis, it is not clear how the qualitative values are used to decide between candidate solutions or how values of PS vs. FS would be incorporated into the use of preferences or options.

Classification: We classify this work similarly to the work of (Giorgini et al., 2004b). Ernst et al. use the backward and forward analysis approaches introduced in by Giorgini et al. The approach also uses a form of local analysis to add optional requirements to candidate solutions. In some ways, human intervention is required in order to understand and select between multiple dominant solutions produced by the algorithm, although this step would occur once answer have been found, and not interactively as part of the procedure.

Table 9 Classification of (Ernst, Mylopoulos, Borgida, & Jureta, 2010)

	Approach							**Analysis Results**			**Additional Notation Supported**			**Analysis Scope**	
Approach	**Satisf Forwds**	**Satisf Backwds**	**Human Interv**	**Metrics**	**Plan-ning**	**Simu-lation**	**Model Check**	**Qual**	**Quant**	**Binary**	**Depend-encies**	**Soft-goals**	**Contrib. Links**	**Global**	**Local**
Ernst et al., 2010	Y	Y	M	N	N	N	N	Y	N	Y	N	M	Y	Y	Y

(i* Evaluation): Previous work has used the qualitative evaluation procedure based on the NFR Framework, and implemented in the Organization Modeling Environment Tool ("OME, Organization Modelling Environment," 2008) in order to perform evaluation on i* models, see (Liu & Yu, 2004; Liu, Yu, & Mylopoulos, 2003; Yu & Cysneiros, 2003) for examples. Such work assumed that the NFR procedure described in (Chung et al., 2000) could be easily extended for use with i*, and did not focus on describing the necessary expansions to the procedure. Horkoff, (2006) has tested this assumption more extensively, using multiple example applications.

Classification: We group this work together as i* Evaluation and classify it in Table 10 as qualitative satisfaction analysis in the forwards direction. These procedures consider the agent-oriented dimension with explicit propagation of evidence across agent boundaries using dependencies. Although analysis of agents internals is supported, this work is classified is as global, as the procedures do not explicitly consider agent boundaries when propagating over them. However, it could be argued that the satisfaction of goals within an actor in a global analysis is a reflection of the local analysis of that actor. In other word, that there is no distinction between global and local analysis, as the model represents both the global and local views working together, and global results provide local results by examining the results per actor. However, because this work does not explicitly allow for local analysis, we keep only the global classification in Table 10.

Table 10 Classification for (i* Evaluation)

	Approach							**Analysis Results**			**Additional Notation Supported**			**Analysis Scope**	
Approach	**Satisf Forwds**	**Satisf Backwds**	**Human Interv**	**Metrics**	**Plan-ning**	**Simu-lation**	**Model Check**	**Qual**	**Quant**	**Binary**	**Depend-encies**	**Soft-goals**	**Contrib. Links**	**Global**	**Local**
i* Evaluation	Y	N	Y	N	N	N	N	Y	N	Y	Y	Y	Y	Y	N

(Horkoff, 2006): In (Horkoff, 2006), the M.Sc. thesis of the author, the procedure described in (Chung et al., 2000) is expanded to work on i* models, taking into consideration dependency links, and additional element types. As the description of the procedure in (Chung et al., 2000) was given only in high-level prose, the work in (Horkoff, 2006) added many details to the procedure, including the treatment of a mixture of links, definition of initial values, propagation from links to links, a consideration of convergence and termination. The conditions for the application of human judgment were relaxed, allowing the user more freedom in their choice. The procedure allows partial values as the end results of evaluation, not encouraging users to promote or demote partial values unless they deem it appropriate as per the domain. The procedure was implemented in a now outdated version of the OpenOME (2010) tool and was tested via application to several example domains, including a consideration of trusted computing technology and a case study involving a not-for-profit institution. We review the case studies used in this work briefly in Chapter 12.

This work considered the benefits of goal model evaluation beyond the analysis of high-level design alternatives. Specifically, this work examined the ability of qualitative i* analysis to provoke manual checks of the syntactic and semantic correctness of model, to provoke further domain elicitation, and using syntactic and semantic discoveries to drive model iteration, improving the overall quality of the model. The framework introduced in this work includes and extends the forward procedure in (Horkoff, 2006), creating a more precise definition of forward propagation, reconsidering convergence and termination, re-implementing the procedure and testing hypothesized benefits through multiple case studies.

Classification: The (Horkoff, 2006) analysis procedure covers the same classification categories as previous work in i* evaluation (i* Evaluation).

(Amyot et al., 2010; International Telecommunication Union, Telelcommuniction Standardization Sector, 2008): Z.151, an International Telecommunication Union, Group 17, Standard describes the User Requirement Notation. URN contains the Goal-oriented Requirement Language (GRL), a variant of i* which relaxes some of the syntax restrictions original to the i* Framework. This work and the work by (Amyot et al., 2010) have introduced several evaluation procedures applicable to GRL models. One evaluation procedure presented in the standard is purely qualitative, and bears similarities to several of the procedures previously described. The procedure uses the same qualitative scale as Chung et al.

(2000), Horkoff (2006), and (i* Evaluation), but uses a slightly modified set of propagation rules, making differing propagation choices. The procedure also uses a different algorithm, propagating values in the order of their link types (first decomposition, then contributions, then dependencies). The most significant difference of this procedure from the procedures in Chung et al. (2000), Horkoff (2006), and (i* Evaluation) is the avoidance of human intervention via a set of rules which automatically determines the values of softgoals in all cases. The number of each type of qualitative contribution towards a softgoals is counted, and, depending on how these numbers compare to each other, a value is determined. This procedure also differs in its treatment of conflicting values, propagating a "none" value when a conflict occurs in order to urge the modeler to resolve the conflict at its source.

In addition to satisfaction levels, GRL contains the ability to assign qualitative or quantitative importance levels to goals. These levels are used in the calculation of an overall satisfaction value for an actor. If an element of an actor has an important value other than none, its value is counted towards the overall satisfaction of an actor, using set rules which are similar to the rules used to decide on final evaluation values for softgoals.

Classification: Work in Z.151 URN (2008) and Amyot et al. (2010) include a description of several analysis procedures applied to GRL models. We classify this work in Table 11 by grouping the capabilities of these procedures together. All procedures introduced in the standard give results in terms of the satisfaction or denial of goals; qualitative, quantitative, and hybrid procedures are described; and only forward propagation procedures are described. Procedures in this work propagate across contribution links, dependency links, and explicitly differentiate between soft and hard goals. Because the satisfaction of agents is calculated, we consider this as a form of local analysis, and classify these procedures as such. We note that the procedure requires importance information in our Table 38 summary.

Table 11 Classification of (Amyot et al., 2010; International Telecommunication Union, Telelcommuniction Standardization Sector, 2008)

	Approach							**Analysis Results**			**Additional Notation Supported**			**Analysis Scope**	
Approach	**Satisf Forwds**	**Satisf Backwds**	**Human Interv**	**Metrics**	**Plan-ning**	**Simu-lation**	**Model Check**	**Qual**	**Quant**	**Binary**	**Depend-encies**	**Soft-goals**	**Contrib. Links**	**Global**	**Local**
Z.151, 2008, Amyot et al., 2010 Pourshahid et al., 2011	Y	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y	Y

Discussion: Although much effort has been placed in the qualitative evaluation of goals, issues and omissions in the approaches can be found. The approaches of Chung et al. (2000) and Giorgini et al. (2002, 2004a, 2004b, 2005) focus only on goal models, without supporting agent-oriented concepts. Furthermore, all approaches reviewed thus far, except for Horkoff (2006) are intended only for the comparison of design alternatives, and do not explicitly address many of the other desiderata for Early Requirements analysis, including improving model completeness or accuracy, encouraging stakeholder involvement, or producing methods which are usable in practice.

Several of the existing qualitative methods which explore goal satisfaction are fully automated, (Giorgini et al. 200*, Amyot et al., 2010), either separating negative or positive evidence or using fixed rules to decide values for softgoals. The first approach often results in the proliferation of conflicts values (Giorgini et al. 200*), while the second approach produces many partial values (Amyot et al., 2010). The proliferation of conflicting or partial values hinders the ability to analyze or compare model alternatives. A combined summary of each of the qualitative analysis procedures can be found in Table 12.

Table 12 Summary of Classifications for Qualitative, Forward Analysis Procedures

	Approach							**Analysis Results**			**Additional Notation Supported**			**Analysis Scope**	
Approach	**Satisf Forwds**	**Satisf Backwds**	**Human Interv**	**Metrics**	**Plan-ning**	**Simu-lation**	**Model Check**	**Qual**	**Quant**	**Binary**	**Depend-encies**	**Soft-goals**	**Contrib. Links**	**Global**	**Local**
Chung et al., 2000	Y	N	Y	N	N	N	N	Y	N	Y	N	Y	Y	Y	N
Giorgini et al., 2002, 2004a	Y	N	N	N	N	N	N	Y	Y	Y	N	M	Y	Y	N
Giorgini et al., 2004b	Y	Y	N	N	N	N	N	Y	N	Y	N	M	Y	Y	N
Giorgini et al., 2005	Y	Y	N	N	N	N	N	Y	N	Y	M	Y	Y	M	Y
Ernst et al., 2010	Y	Y	M	N	N	N	N	Y	N	Y	N	M	Y	Y	Y
i* Evaluation, Horkoff, 2006	Y	N	Y	N	N	N	N	Y	N	Y	Y	Y	Y	Y	N
Z.151, 2008, Amyot et al., 2010, Pourshahid et al., 2011	Y	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y	Y

2.3.1.3 Quantitative Satisfaction Analysis

Several approaches calculate the satisfaction and denial of goals using quantitative measures.

(Giorgini, Mylopoulos, Nicchiarelli, & Sebastiani, 2002; 2004b): The qualitative procedure described in (Giorgini et al., 2002; 2004a) is adapted to produce a quantitative version in the same work. In order to propagate quantitative values, the goal model contribution links must be adjusted to contain numerical weights. As with the qualitative version, positive and negative evidence is stored separately. Goals are again given Sat and Den variables, where Sat(G) = c means that there is at least c evidence of Sat(G). Here the c values range over a numerical interval [inf, sup], where inf represents no evidence and sup represents full evidence. In the examples, a range of [0,1] is used, both for the satisfaction and denial of goals as well as the weights of contribution links. The rules are adjusted to deal with these numerical values via the introduction of the $\oplus$ operator, used as disjunction or "max", and the operator $\otimes$, used as conjunction or "min". The $\otimes$ operator is defined as typical multiplication. The $\oplus$ operator is defined as follows:

$$x \oplus y = x + y - x \times y$$

In this scheme, the results of contributions indicate the conditional probability of the parent goal being satisfied, given the satisfaction of the child goal. The application of this numerical model to a goal graph creates a Bayesian network.

The propagation rules for this method are consistent with those of the qualitative version. In AND links positive values are combined via conjunction and negative values are combined via disjunction. The reverse holds for OR links. Evidence propagated through partial links (+, -) is combined via conjunction with the numerical strength of the links. Evidence through full links is propagated without change. When combining multiple sources of evidence with the same polarity (all positive/negative) to a single goal, the maximum value is taken. However, the procedure does not promote partial (PS/PD) values to full values, even if multiple sources of evidence are present, making the results not cumulative.

In this algorithm, termination is reached when the old and new values for all goals in the model differ only by a very small constant. As the values are non-decreasing, the values are guaranteed to converge, and the algorithm terminates.

Giorgini et al. describe a qualitative and quantitative forward analysis procedure in (2002, 2004a), while in (Giorgini et al., 2004b) the authors describe a backward procedure focusing on qualitative propagation. However, the backward quantitative procedure has been implemented in the GR-Tool (GR-Tool, 2010).

This work has been classified in the previous section.

Several pieces of work in the Tropos family of approaches have used and adapted the evaluation procedures introduced by (Giorgini et al., 2002; 2004a; Giorgini et al., 2004b). For example, work (Asnar et al., 2007) applies Tropos modeling and analysis to compare design alternatives in a case study involving E-Voting.

(Asnar & Giorgini, 2006): Asnar & Giorgini (2006) use Tropos analysis procedures to find the best candidate design solution, taking risk into account. The approach argues that typical risk measurement approaches, such as Fault Tree Analysis, fail to analyze risks for the organizational setting of the system, especially vital for safety critical systems. In this work, they propose the Goal-Risk Model Framework, an extension of the Tropos Framework which includes events, risks, which can influence the satisfaction of goals, and treatments, which can mitigate events. The constructs are divided into separate model layers: a goal layer, event layer and treatment layer. In this case Sat or Den values represent the likelihood of an event occurring or the success-rate of a countermeasure, and contribution links representing the effects of events on goals and the effects of countermeasures on events.

They evaluate the model and find the best solution in steps, using the approach by (Giorgini et al., 2004b). First they apply backwards analysis on the goal model layer, identifying candidate solutions in light of their target goals. Then they identify acceptable levels of risk by placing thresholds of Den values on the goals in the model. Forward analysis as described in (Giorgini et al., 2002; 2004a) is used to propagate the effects of events, determining which candidate solutions have acceptable levels of risk. Candidate solutions with non-acceptable levels of risk are analyzed again in light of potential combinations of countermeasures, determining whether the risks can be acceptably reduced. Finally, all candidate solutions with sufficiently low levels of risk are evaluated against minimum cost criteria, as is done by (Giorgini et al., 2004a), to find the most-inexpensive candidate solution. This work goes further and categorizes various treatments, although this does not seem to effect the application of the analysis procedure. The framework has been implemented in GR-Tool, as is used Tropos evaluation. Asnar et al. (2007) expand the 2006 approach to work with an agent-oriented viewpoint and to utilize a planning approach, discussed further in Section 2.3.3.

Classification: We classify (Asnar & Giorgini, 2006) as using satisfaction analysis. This approach could be thought to include metrics; however, given that the metric used to calculate risk is the Den value in Satisfaction analysis, the use of metrics is debatable. Analysis using metrics will be discussed further in Section 2.3.2. The method introduced by Asnar & Giorgini refers to work in (Giorgini et al., 2004b), but it is not clear whether it applies both qualitative and quantitative analysis, represented with an "M" in Table 13. It does, however, use both forwards and backwards evaluation. This particular method uses the supplementary information of events, treatments, and cost, summarized in Table 38.

Table 13 Classification for (Asnar & Giorgini, 2006):

	Approach							Analysis Results			Additional Notation Supported			Analysis Scope	
Approach	**Satisf Forwds**	**Satisf Backwds**	**Human Interv**	**Metrics**	**Plan-ning**	**Simu-lation**	**Model Check**	**Qual**	**Quant**	**Binary**	**Depend-encies**	**Soft-goals**	**Contrib. Links**	**Global**	**Local**
Asnar & Giorgini, 2006	Y	Y	N	N	N	N	N	Y	M	Y	N	M	Y	Y	N

(Amyot et al., 2010; International Telecommunication Union, Telelcommuniction Standardization Sector, 2008): Several evaluation procedures for GRL models were introduced in (International Telecommunication Union, Telelcommuniction Standardization Sector, 2008)**,** the URN standard, with further detail in (Amyot et al., 2010). The qualitative methods of this work were reviewed in the previous section. Quantitative and hybrid propagation procedures are also defined, expanded from an earlier quantitative procedure described by (Roy, Kealey, & Amyot, 2006). The quantitative propagation uses propagation rules similar to those used by Giorgini et al., propagating numbers from -1 to 1 through decomposition links, dependency links, and contribution links, the latter of which are annotated with numbers in the range -100 to 100. Resulting values are summarized using the qualitative notations of the NFR Framework (Chung et al., 2000). This method uses a form of summation and normalization to calculate the evaluation results of softgoals automatically. Thresholds are used to prevent elements receiving full (-1 or 1) values via summation unless a full value is received. Positive and negative numbers are summed at each propagation step, automatically resolving conflicts. An assessment of overall actor satisfaction is produced, factoring the numerical priority and criticality of each actor's elements.

A hybrid algorithm is also introduced, propagating quantitative values through qualitative contribution links which are converted to a set value, for example, the value 25 is used for Help links.

This approach defines useful categories by which to sort evaluation algorithms applied to GRL models, namely the inclusion or treatment of: actor satisfaction, automation, cycles, conflict, strategy consistency (allowing inconsistent initial values), evaluation overriding (whether initial values can be overridden by subsequent propagation), relation to UCM (only relevant for GRL), evaluation ordering for links (through which links the propagation occurs in what order), link evaluations (rules for propagation), and tolerance. We have incorporated some of the higher-level qualities which have applicability beyond GRL into our categorization table, for example, actor satisfaction is considered in the global vs. local scope, and automation is considered in the consideration of human judgment. Other categories such as strategy consistency and evaluation overriding are judged to be too detailed to consider for each approach.

An extension to this work by (Pourshahid et al., 2008) uses quantitative Key Performance Indicators (KPI) derived from concrete business process measurements to feed values into the GRL models, which are then propagated using the algorithms described by (Roy, Kealey, & Amyot, 2006) in order to assess the satisfaction of goals.

These methods have been categorized in the previous section.

(Pourshahid, Richards, & Amyot, 2011): Work in (Amyot et al., 2010) and (Pourshahid et al., 2008) is expanded with a focus on enabling decision making. The work argues that current business intelligence systems are insufficient: the data models do not match what is needed for decisions, they do not show key relationships between data and goals, and the visualization of data is poor. They adapt the GRL goal framework to be used as a business intelligence decision-making framework. They argue that this better enables managers to make decisions by making goals and relationships explicit, working over limited data, and providing visual support. Specifically, work in (Pourshahid et al., 2008) using KPIs with GRL is expanded to allow for formula-based goal evaluation, where domain specific formulas allow the propagation of KPI values in a forward direction in a GRL model. For example, a Profit task can be calculated by the value costs and stolen items tasks subtracted from a revenue task. Although each formula must be manually added to the model, the model does not require all propagation to be done via formula-based evaluation. Values from the formulas are fed into quantitative values as from (Amyot et al., 2010) by using a normalization function which requires the threshold, target, and current value. In this way, the method allows for analysis over partial, detailed domain information. The method also introduced a methodology for model construction and KPI elicitation, including a consideration of risk in the final stage. The approach is tested on a case study involving a retail business in Ottawa.

This approach is classfied in the same categories as (Amyot et al., 2010).

(Barone, Jian, Amyot, & Mylopoulos, 2011): Barone et al. (2011) focus on analysis of strategic business models drawn using the BIM framework as defined in (Barone, E. Yu, Won, Jiang, & Mylopoulos, 2010). Similar to the work in (Pourshahid et al., 2011) , KPIs are integrated with models containing goals and relationships for the purpose of analysis derived from both goals and data collected as part of business operations. This work focuses on the definition of composite indicators, indicators whose values are defined by their components, which may also be indicators. The aim of the work is to perform propagation over models using indicator values and values derived from composite indicators. Reasoning with indicators can used mixed approaches depending on information available. Propagation can use domain specific indicators, specific formula, unit conversion, and quantitative or qualitative goal model propagation, as is described in (Giorgini et al., 2004b).

The work outlines a classification of analysis techniques over BI models, with techniques ranging from quantitative (accurate), quantitative (heuristic), quantitative (normalized), and qualitative. These techniques are gradually less accurate from first to last, respectively. In the first technique values are calculated from business intelligence data using accurate mathematical functions. In the second technique, these calculations involve some sort of conversion, for example from hours to dollars. The third technique uses a normalization function to convert between indicators and quantitative goal analysis values, similar to (Pourshahid et al., 2011). The last technique maps indicator values to qualitative goal satisfaction values, as used in (Giorgini et al., 2004b). Using these techniques, depending on the amount of detailed information available, a combination of analysis approaches can be applied to the same model at once.

This technique uses forward quantitative and qualitative analysis. The BIM framework supports contribution links but not dependencies. It includes only a "goal" concept, and not explicitly a softgoal, but similar to (Giorgini et al., 2004b), goals can take on partial analysis values, taking on characteristics of softgoals. Analysis is performed globally over the model.

Table 14 Classification of (Barone et al., 2011)

	Approach							Analysis Results			Additional Notation Supported			Analysis Scope	
Approach	**Satisf Forwds**	**Satisf Backwds**	**Human Interv**	**Metrics**	**Plan-ning**	**Simu-lation**	**Model Check**	**Qual**	**Quant**	**Binary**	**Depend-encies**	**Soft-goals**	**Contrib. Links**	**Global**	**Local**
Barone et al., 2011	Y	N	N	N	N	N	N	Y	Y	Y	N	M	Y	Y	N

(Letier & van Lamsweerde, 2004): Methods have been proposed in order to evaluate the satisfaction of goals in the KAOS Framework (Dardenne et al., 1993). The KAOS Framework aims to represent the domain and its requirements focusing on intentions, but in a formal way, where all goals are eventually refined to concrete, formal goals. Work by Letier and van Lamsweerde introduces the notion of partial satisfaction for goals in KAOS via the creation of a probabilistic model, where the satisfaction level of a goal corresponds to the likelihood of its satisfaction, given concrete domain evidence. This method requires the definition of specific cumulative distribution functions, called objective functions, over random variables, called quality variables, for goals in the model. Quality variables for leaf goals must be probabilistically independent. Objective functions have a mode indicating whether they should be minimized or maximized, a target probability, and a current probability. These functions do not need to be defined for each goal in the model, as the quality variables of a parent goal are related to quality variables of sub-goals using domain specific refinement equations. Propagation occurs when distribution estimations are found for leaf goals and refinement equations are reformulated into probability density functions in order to compute objective functions. The measure of partial satisfaction for a goal is equivalent to the probabilities of the objective functions. This method works in both a forwards and backwards direction.

This approach guides the elaboration of models using heuristics to identify quality variables and objective functions. This process of refinement may reveal problems in the current system design.

Classification: We classify the work by Letier & van Lamsweerde as measuring satisfaction, although satisfaction in this case means the probability of accomplishment. The method is quantitative, uses forward and backward propagation and takes a global stance. The method is agent-oriented in so much as KAOS is agent-oriented, where leaf-level goals are assigned to agents. In Table 38, we note that this method requires the additional information, namely objective functions, quality variables, and domain refinement equations, summarized as Probabilistic Information.

Table 15 Classification of (Letier & van Lamsweerde, 2004)

	Approach							Analysis Results			Additional Notation Supported			Analysis Scope	
Approach	**Satisf Forwds**	**Satisf Backwds**	**Human Interv**	**Metrics**	**Plan-ning**	**Simu-lation**	**Model Check**	**Qual**	**Quant**	**Binary**	**Depend-encies**	**Soft-goals**	**Contrib. Links**	**Global**	**Local**
Letier & van Lamsweerde, 2004	Y	Y	N	N	N	N	N	N	Y	Y	M	N	N	Y	N

(van Lamsweerde, 2009): In response to perceived limitations in the NFR qualitative analysis procedure, van Lamsweerde introduces a quantitative analysis procedure over goal models. Similar to the NFR procedure, this procedure aims to be lightweight, differentiating it from the previous analysis procedure introduced by Letier & van Lamsweerde (2004).

This quantitative procedure supports prioritization by adding different weights to leaf goals, scoring each alternative against leaf goals using measures from the domain, and collecting analysis results in a matrix for comparison. Results from this table can be treated as rough estimates for how well each alternative scores over the leaf goals in the model. The procedure assigns one or more gauge variables, i.e. real metrics such as time, cost, or quanity, to softgoals in the model. These variables must be able to be propagated cummulatively, and are thus propagated up the goal tree. Gauge results for root goals are assigned maximum and target values. Final scores for each alternative are calculated using the importance of each leaf goal, the score of each option, and how closely each option produces the root target value for its gauge. In other words, the rough estimates using the first two values are adjusted using root target values. The authors argue that the resulting values are more reliable compared to qualitative analysis as they rely on system phenomena and are derived systematically from leaf softgoals. The procedure is illustrated using a simple example.

Although the procedure introduced by van Lamsweerde is simpler (more lightweight) than the procedure introduced in (Letier & van Lamsweerde, 2004), it also comes with some limitations. Although grounding the gauge measures in system phenomena is likely to increase the accuracy of results, it is still questionable how difficult it is to gather such information in the domain, especially in early analysis stages, and especially if the gauges and the model are aiming for completeness. The procedure avoids the challenge of merging dissimilar measures (time, cost, etc.) by propagating these values up the graph separately. However, if values are incomplete, if not all leaf goals propagate the same gauge, it is difficult to calculate gauge results over AND relationships. It seems that in this case, gauge values are ignored. The procedure emphasizes the value of alternatives over leaf level goals, when in fact, it is often root goals which are the most critical. When collecting scores for alternatives against leaf goals, the numerical values are estimates from domain experts. However, it may be difficult to come up with accurate and reliable quantitative estimates, as argued in (Elahi & E. Yu, 2011). The procedure does not describe propagation over contribution links, only AND/OR links. Finally, it is not clear how multiple gauge variables per leaf goal are factored into the final scores for each alternative, or how the procedure would deal with analysis over multiple root goals.

Classification: We classify the work by van Lamsweerde as a forward analysis procedure, facilitating binary and quantitative analysis. It supports softgoals and supports qualitative contributions of alternatives to leaf softgoals. The procedure does not consider dependencies or support local analysis.

Table 16 Classification of (van Lamsweerde, 2009)

	Approach							Analysis Results			Additional Supported Notation			Analysis Scope	
Approach	**Satisf Forwds**	**Satisf Backwds**	**Human Interv**	**Metrics**	**Plan-ning**	**Simu-lation**	**Model Check**	**Qual**	**Quant**	**Binary**	**Depend-encies**	**Soft-goals**	**Contrib. Links**	**Global**	**Local**
van Lamsweerde, 2009	Y	N	N	N	N	N	N	N	Y	Y	N	Y	M	Y	N

Discussion: Although quantitative evaluation can be useful for a finer-grained analysis, approaches to this type of analysis have potential issues in their analysis capabilities. The quantitative approaches by Giorgini et al. contain a separation of positive and negative evidence, making an overall judgment of element satisfaction and design alternative effectiveness difficult. The GRL approaches use of tolerance seems restrictive, although this value can be set such that it is ignored. The conversion of qualitative to arbitrary quantitative numbers in the hybrid GRL approach can be problematic, as additional information is inferred where none was present.

In general, when examining quantitative goal model analysis methods, unless the numbers used in the evaluation are derived directly from specific measures in the domain, they can be viewed as a form of fine-grained qualitative analysis. In quantitative approaches, instead of prompting for expert judgment during the analysis, judgment is pre-added to the model through numerical values, often attached to contribution links or initial analysis values. An expert may say that the element has a satisfaction value of 0.7/1.0 or 30/100, or these values may be taken from measurements in the domain, but when these values are propagated throughout a model, the associated level of approximation increases. In this case there is a danger that analysts, particularly stakeholders who are not familiar with goal modeling, may place an undeserved amount of confidence in the analysis results, as numerical results are often associated with mathematical precision.

Methods by Pourshahid et al. and Barone et al. allow the addition of incomplete KPIs and domain specific indicators, formula, and composite indicators to goal model analysis. This mix of approaches allows for use of specific domain information while acknolwding the incomplete and imprecise nature of early requirements analysis. Barone et al. acknowledge that the accuracy of analysis descreases as the precision of formula and conversion decreases. Although these approaches are a good first step in allowing more precise domain information to be integrated with early RE considerations, the conversions and resulting quantitative values are still likely to be highly approximate, suffering from the same issues as other quantitative work in this area.

Table 17 summarizes both qualitative and quantitative satisfaction analysis procedures.

Table 17: Summary of Satisfaction Propagation Goal Model Analysis Procedures

	Approach							Analysis Results			Additional Notation Supported			Analysis Scope	
Approach	**Satisf Forwds**	**Satisf Backwds**	**Human Interv**	**Metrics**	**Plan-ning**	**Simu-lation**	**Model Check**	**Qual**	**Quant**	**Binary**	**Depend-encies**	**Soft-goals**	**Contrib. Links**	**Global**	**Local**
Maiden et al., 2007	Y	N	Y	N	N	N	N	N	N	Y	Y	Y	Y	Y	N
Jureta et al., 2008, 2010	N	Y	N	N	N	N	N	N	N	Y	N	Y	N	Y	N
Chung et al., 2000	Y	N	Y	N	N	N	N	Y	N	Y	N	Y	Y	Y	N
Giorgini et al., 2002, 2004a	Y	N	N	N	N	N	N	Y	Y	Y	N	M	Y	Y	N
Giorgini et al., 2004b	Y	Y	N	N	N	N	N	Y	N	Y	N	M	Y	Y	N
Giorgini et al., 2005	Y	Y	N	N	N	N	N	Y	N	Y	M	Y	Y	M	Y
i* Evaluation, Horkoff, 2006	Y	N	Y	N	N	N	N	Y	N	Y	Y	Y	Y	Y	N
Z.151, 2008, Amyot et al., 2010, Pourshahid et al., 2011	Y	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y	Y
Asnar & Giorgini, 2006	Y	Y	N	N	N	N	N	Y	M	Y	N	M	Y	Y	N
Barone et al., 2011	Y	N	N	N	N	N	N	Y	Y	Y	N	M	Y	Y	N

Letier & van Lamsweerde, 2004	Y	Y	N	N	N	N	N	N	Y	Y	M	N	N	Y	N
van Lamsweerde, 2009	Y	N	N	N	N	N	N	N	Y	Y	N	Y	M	Y	N

2.3.2 Analysis using Metrics

Several approaches to the analysis of goal and agent-oriented models apply metrics to goal models to measure specific qualities over the model other than general goal satisfaction.

(Franch & Maiden, 2003): Franch & Maiden (2003) apply metrics to i* SD models, built to support the selection of COTS options for system development. They describe their approach using a meeting scheduler example, producing SD and limited SR models of the domain. They derive six system properties that can be measured over the SD models: Diversity, Vulnerability, Packaging, Self-Containment, Uniformity, and Connectivity. These properties are precisely defined using formulas over counts of various classifications of dependencies. They distinguish between instance and model dependencies, where there can be multiple instance dependencies for a single model dependency. They also distinguish between duplicated and non-duplicated model dependencies, hidden and non-hidden dependencies, resource and non-resource dependencies, dependencies from Users or External Agents, and between Components interacting with each type of agent. Generally the differences between these types of dependencies are not well defined. Hidden dependencies seem to be Instance dependencies between agents who are implemented by the same component, although it is not clear why this is "hidden".

Properties are calculated based on the number of occurrences of these types of dependencies. For example, to calculate Diversity, they use: (Instance Dependencies – Model Dependencies)/Instance Dependencies. This number is larger when there are more duplicate dependencies, and the system is more Diverse and reliable. In another example, Packaging, the grouping of system characteristics into components, is calculated by the number of Hidden Instance Resource Dependencies/Instance Resource Dependencies, with the idea that when a component packages more functionality, dependencies between actors implemented by these components are implicitly satisfied. Here, a higher number means more of these implicitly satisfied components.

They close the paper by describing how to select possible architecture instances based on model restrictions, and how to select components in light of existing "anchors" (pre-existing system and constraints) in the domain.

Classification: This approach applies quantitative metrics over i* models. For this type of work, we use the Metrics column of the Approach section in Table 18. The metrics are global, with the entire approach considering agent-oriented models. This approach does not incorporate extra information into models, but classifies dependencies and actors in terms of their relationships to COTS components.

Table 18: Classification of (Franch & Maiden, 2003):

	Approach							Analysis Results			Additional Notation Supported			Analysis Scope	
Approach	**Satisf Forwds**	**Satisf Backwds**	**Human Interv**	**Metrics**	**Plan-ning**	**Simu-lation**	**Model Check**	**Qual**	**Quant**	**Binary**	**Depend-encies**	**Soft-goals**	**Contrib. Links**	**Global**	**Local**
Franch & Maiden, 2003	N	N	N	Y	N	N	N	N	Y	N	Y	Y	N	Y	N

(Franch, Grau, & Quer, 2004): Franch et al. focus on developing structure analysis metrics for Actor-Dependency models, (in i*, Strategic Dependency Models). Their aim is to use such metrics to analyze non-function and organizational goals over the models. The use of SD models in this work includes attributes or properties for actors and dependencies, such as priorities or importance. Actors and dependencies are classified into sorts, such as human or computer, goal or task, etc.

The approach introduces a generic framework with three different categories of metrics, aiming to support metric extensibility. Each category of metric relies on actor and dependency evaluation. In actor evaluation, an actor is evaluated for a specific property by multiplying its weight by a function based on its incoming or outgoing dependencies. Given a property, a dependency evaluation is the product of a dependency weight and a function involving the depender and dependee actors.

The first type of structural measure is global over the model as a whole. These are calculated either for all actors or all dependencies in the model, summing the values for a property and optionally normalizing by a varying factor. Local structural measures are calculated in the same way, but taking the sum only for a particular set of actors, or for the dependencies of particular actors. Sensitivity measures act as a summary by providing the mean value of a local structural measure.

They demonstrate their approach with a meeting scheduler example. In the example, design alternatives are evaluated in terms of privacy (data privacy), accuracy (data accuracy), efficiency (process agility), and fault tolerance (responsibility dissolution), defining structural metrics for each concern. The first three metrics are global, dependency-based metrics. The type of dependency and the type of dependees and dependers (human or software, role or agent) is examined, and each combination is assigned a value between 0 and 1 for each of the first three metrics. These values are summed per metric and the final values are calculated by adding a normalizing factor which varies per metric: two are not normalized while one is normalized by the number of resource dependencies. The last metric, responsibility dissolution, is defined as a sensitivity actor based metric. For this metric, for a particular actor, incoming dependencies are counted and then divided by the total number of dependencies. The resulting metrics are analyzed and a system alternative, an automated scheduler, is recommended.

The case study is extended to consider some COTS-based alternatives for the automatic meeting scheduler. Architectural alternatives are evaluated by a metric concerning the complexity of the user interface. This involves a count of the interactions between COTS components and human actors, and adds more detail by considering dependency types and weights. Either qualitative or quantitative measures can be used in the weighting scheme.

Classification: This approach is metrics based, using structural metrics over the model. Although the approach is mainly quantitative, the use of qualitative importance measure is mentioned. Although the metrics are defined manually for each model, at least until a catalogue of metrics can be produced, the calculation of metrics is automatic. Both global and local metrics are supported. This approach incorporates additional information by including actor and dependency weights or importance values, as reflected in Table 38.

Table 19: Classification of (Franch et al., 2004)

	Approach							**Analysis Results**			**Additional Notation Supported**			**Analysis Scope**	
Approach	**Satisf Forwds**	**Satisf Backwds**	**Human Interv**	**Metrics**	**Plan-ning**	**Simu-lation**	**Model Check**	**Qual**	**Quant**	**Binary**	**Depend-encies**	**Soft-goals**	**Contrib. Links**	**Global**	**Local**
Franch et al., 2004	N	N	N	Y	N	N	N	M	Y	N	Y	Y	N	Y	Y

(Franch, 2006): Franch continues the work in Franch & Maiden (2003) and Franch et al., (2004) and outlines a more general framework which uses the structure of i* models as a means to measure desired properties such as security, completeness, modifiability and predictability. This general framework calls for the development of individual metrics over all elements/actors or subsets of models which return values in the form of numbers, Booleans, or sets of i* elements/actors. Properties can be quantitative or qualitative, can require expert judgment, and are expressed in OCL. Metrics can be built up from smaller metrics.

As an example metric, one could measure the predictability of dependencies. Here they suggest that task and resource dependencies are totally predictable, based on the definitions of these concepts by Yu (1997), but that goal and softgoal dependencies have varying predictability. They give three options for finding goal and softgoal dependency predictability: assigning fixed weights to all such dependencies, assigning variable weights to such dependencies by expert judgment, or by finding a quantitative measure for predictability. They focus on the last option, defining a metric in terms of depender expertise and know-how, with know-how for softgoal dependencies defined over the number of dependees contributing to the dependum, the more contributions, the more predictable the dependency is. In this case, the measure of predictability seems to correspond better to reliability. For goal dependencies, predictability is decreased when there are more decomposition combinations available to satisfy a goal. They continue by suggesting metrics to measure the predictability of actors, predictability of scenario paths, and predictability of the entire model.

Classification: We classify this work as using metrics and apply both qualitative and quantitative analysis. This approach can use human intervention, works on agent-oriented models and can provide both local and global metrics.

Table 20 Classification for (Franch, 2006)

	Approach							Analysis Results			Additional Notation Supported			Analysis Scope	
Approach	**Satisf Forwds**	**Satisf Backwds**	**Human Interv**	**Metrics**	**Plan-ning**	**Simu-lation**	**Model Check**	**Qual**	**Quant**	**Binary**	**Depend-encies**	**Soft-goals**	**Contrib. Links**	**Global**	**Local**
Franch, 2006	N	N	Y	Y	N	N	N	Y	Y	N	Y	Y	Y	Y	Y

The metrics approach introduced by Franch et al. (2004) and Franch (2006) are applied by (Grau & Franch, 2007) to evaluate the effectiveness of alternative architectures discovered via a systematic process. In this work, metrics are derived using the Goal Question Metric (GQM), used to produce statements about the evaluation of software architectures in terms of quality attributes. These questions are treated as goals, finding questions to evaluate them and then using metrics to evaluate these questions.

This work also applies metrics for computing the cohesion and coupling of i* models. Coupling is measured by the number of actors an actor is associated with while cohesion is a measure of the repetition of the number of dependencies that start from or go through each actor. These measures appear to be problematic. First, it is not clear how these measures accurately reflect cohesion and coupling, especially the measure of cohesion. Second, it is difficult to determine the meaning of these concepts for i* models. For classes and general system design, coupling is clearly undesirable, while cohesion is desirable. These assumptions may not hold as well for system actors. Coupling may be positive in a social situation if it involves offloading many responsibilities, while cohesion could be negative if an actor values diversity or variety.

Metrics are also used in PRiM (Process Reengineering i* Method), described by (Grau et al., 2008), based on the work by (Franch & Maiden, 2003) and (Franch et al., 2004). Here they divide the application of metrics into four steps. First, the selection of suitable properties to be evaluated over the models, these are identified as the most important NFRs. Actor or dependency based metrics are then defined for these properties. Next, they evaluate the alternative i* models in terms of these metrics, here they choose to evaluate metrics for ease of communication and process agility. Finally, they perform trade-off analysis over the metric results, comparing results to find the best alternative. If the comparison of alternatives is not clear, they recommend iteration, considering new metrics, and refining existing metrics.

(Kaiya et al., 2002): Work by Kaiya et al. (2002) introduces the AGORA approach, Annotated Goal-Oriented Requirements Analysis. This approach attempts to address some of the missing capabilities of existing goal-oriented approaches, including a consideration of goal priorities, solving goal conflicts, selecting alternatives, and measuring the quality of models. The procedure takes a basic AND/OR goal graph and annotates the graph with various information: preferences attached to goals, contribution values attached to AND/OR edges and rationale attached to nodes and edges, explaining the reasoning behind the constructs. Contribution values are expressed as integers between -10 and 10; differing values can be placed on OR edges, but the same values should be placed on all AND edges for a parent goal, as all AND decomposition elements are required. Stakeholders give priorities to goals using the same scale. Priorities are stored in matrices, as stakeholders provide not only their own priorities but an estimate for the priorities of other stakeholders. Differences in these values are analyzed to find divergent opinions concerning the domain.

Contributions and preference values are used to help the analyst decide which alternative to select in an OR decomposition. These values are not propagated, but are analyzed in an ad-hoc way. Conflicts amongst contributions or preference values are resolved via further decomposition or stakeholder negotiation.

The approach by Kaiya et al. applies quality metrics over its annotated AND/OR goal trees. It mentions the existence of several metrics, and provides specific examples to measure correctness, unambiguity, completeness, consistency, verifiability, modifiability, and traceability. They define metrics for correctness in terms of contribution values in the paths to the customer's goals, and in terms of the customer's priorities, with the idea that correctness is how closely the model meets the customer's needs. Unambiguity is measured by looking for ambiguous phrases in goals, and by examining the gap between stakeholder's prioritizations. Completeness is a measure of how many of the customers goals are contributed to positively. Consistency is related to conflicts among goals, being measured in terms of positive and negative contributions, as well as large variances in the preference matrices. Verifiability is measured through the ability to create test cases for final goals. Modifiability is measured by the number of incoming edges to a goal, as that goal would have a significant effect on the model if modified. Traceability is the distance between customer's needs and leaf goals.

Much of the additional information added to the AND/OR graph in AGORA models seems semantically equivalent to constructs in other approaches. Adding contributions to AND/OR decompositions is similar to the use of contribution links from alternatives to softgoals in i* and other goal approaches. The difference is that in the typical goal approaches, the softgoal destination of these links provides information on why an alternative may be positive or negative, i.e., this option hurts maintainability, while in the AGORA approach, this type of information is captured in the textual rationale. The approach does not take full advantage of its use of quantitative contributions by propagating these values through the graph.

Classification: The approach in (Kaiya et al., 2002) is quantitative, measuring a form of partial satisfaction, although it does not explicitly use softgoals. This work does not have a propagation algorithm, but evaluates quality metrics over the model. These metrics are global in nature. The approach does not directly include contribution links, but can include some of this information via textual rationale. The procedure considers agents in the elicitation of priorities, but does not represent dependencies between agents in the model. This approach incorporates additional information in the form of preference matrices and rationales.

Table 21: Classification for (Kaiya et al., 2002)

	Approach							Analysis Results			Additional Notation Supported			Analysis Scope	
Approach	**Satisf Forwds**	**Satisf Backwds**	**Human Interv**	**Metrics**	**Plan-ning**	**Simu-lation**	**Model Check**	**Qual**	**Quant**	**Binary**	**Depend-encies**	**Soft-goals**	**Contrib. Links**	**Global**	**Local**
Kaiya et al, 2002	N	N	N	Y	N	N	N	N	Y	Y	N	N	M	Y	N

(Tanabe et al., 2008). The approach in (Kaiya et al., 2002) for the AGORA framework is expanded in (Tanabe et al., 2008). The impact analysis procedure described in (Tanabe et al., 2008) is focused on change management, detecting conflicts when a new goal is added and analyzing goal achievement when a goal is deleted. When a goal is added, the procedure uses goal characteristics such as security or usability to suggest conflicts between goals. When a goal is deleted the approach calculates impact on the parent goal using a ratio of the contribution values assigned to the links. Unlike in the selected procedures, this value is not propagated further up the graph. The AGORA procedure can also calculate achieve and obstruct values for the roots goals in the graph.

Classification: We classify this approach similarly to the approach in (Kaiya et al., 2002). Although the procedure uses metrics to calculate conflicts and impacts, it also uses a procedure similar to Tropos quantitative evaluation to calculate achieve and obstruct values. We add forward satisfaction to the list of classifications.

Table 22: Classification for (Tanabe et al., 2008)

	Approach							Analysis Results			Additional Notation Supported			Analysis Scope	
Approach	**Satisf Forwds**	**Satisf Backwds**	**Human Interv**	**Metrics**	**Plan-ning**	**Simu-lation**	**Model Check**	**Qual**	**Quant**	**Binary**	**Depend-encies**	**Soft-goals**	**Contrib. Links**	**Global**	**Local**
Tanabe et al., 2008	Y	N	N	Y	N	N	N	N	Y	Y	N	N	M	Y	N

Discussion: Although the application of metrics to the analysis of goal and dependency models can be useful for general qualitative aspects which are effected by many aspects in the model, it is still useful to explicitly represent such aspects as model elements, for example a Predictability softgoal. This is especially true for domain or actor specific qualitative requirements which may be difficult to measure via model structure, such as Employee Satisfaction or Efficient Transaction Process. In addition, meaning derived from the structure of i* models may be inaccurate, especially if model creators were not aware of these potential interpretations when constructing the model. For example, the presence of multiple ways to satisfy a goal may indicate the possibility of high unpredictability, but only if all alternatives have equal likelihood. Often, multiple alternatives are included in an attempt to fully explore potential solutions in the design space, and not to indicate the presence of various run-time solutions. Little thought is given to the accuracy and trustworthiness of these measures. Generally, the semi-formal nature of i* constructs, and the variance in i* styles, makes it difficult place a high degree of confidence in metrics based on the structures of i* models. The approach by Franch (2006) addresses this issue by suggesting that all models are built using the same methodology, suggesting the methodology described by (Grau et al., 2005); however, this can be quite restrictive, as various applications may want to deviate from this methodology while still applying analysis with metrics. Despite this, such metrics may be useful or interesting for a high-level, human-centered, exploration of the model, especially if catalogues of useful and reliable metrics are provided. The techniques applying metrics are summarized in Table 23.

Table 23: Summary of Classifications for Metric Procedures

	Approach							**Analysis Results**			**Additional Notation Supported**			**Analysis Scope**	
Approach	**Satisf Forwds**	**Satisf Backwds**	**Human Interv**	**Metrics**	**Plan-ning**	**Simu-lation**	**Model Check**	**Qual**	**Quant**	**Binary**	**Depend-encies**	**Soft-goals**	**Contrib. Links**	**Global**	**Local**
Franch & Maiden, 2003	N	N	N	Y	N	N	N	N	Y	N	Y	Y	N	Y	N
Franch et al., 2004	N	N	N	Y	N	N	N	M	Y	N	Y	Y	N	Y	Y
Franch, 2006	N	N	Y	Y	N	N	N	Y	Y	N	Y	Y	Y	Y	Y
Kaiya et al, 2002	N	N	N	Y	N	N	N	N	Y	Y	N	N	M	Y	N
Tanabe et al., 2008	Y	N	N	Y	N	N	N	N	Y	Y	N	N	M	Y	N

2.3.3 Planning

Methods have been proposed which apply planning and simulation to the problem of finding satisfactory design alternatives as expressed in agent-goal models.

(Bryl, Giorgini, & Mylopoulos, 2006a): Bryl et al. (2006a) propose a framework using a planner to find satisfactory delegations within a social network represented in simplified i* syntax. The modeler must specify axioms to define how goals can be decomposed, which actors can satisfy which goals, or which types of goals, where type is a domain specific type, and which actors can depend on which other actors. An iterative procedure is used to automatically find a plan, or sequence of actions possibly including delegations, which satisfies the goals of all actors, and then to evaluate that plan in terms of the cost for each individual actor. The procedure repeats until a satisfactory, but not necessarily optimal, plan is found.

This method does not make use of the i* concept of softgoals, but suggests that rules which capture desired non-functional requirements can be integrated into the global criteria for a plan. This method suggests a global evaluation of candidate plans using plan length, plan cost, and non-functional criteria. In this work, the plans are evaluated locally, per actor, in terms of plan cost. The cost of satisfaction, delegation and refinement of goals by an actor are taken into account. If the cost per actor is greater than a cost bound, the authors find the most expensive action of the actor with the lowest cost and try to negate this action when finding a new plan. The algorithm stops when a plan which is under the cost bound for all actors is found. This plan is not necessarily an optimal plan. Satisfaction in this method is treated as a Boolean, not allowing an analysis of the degree of satisfaction. The method is implemented in the P-Tool, similar to the GR tool used by (Giorgini et al., 2004a, 2004b).

Classification: This approach works with the satisfaction of elements to find a plan, although this satisfaction is measured in a binary manner. It also applies a metric for the cost of a plan to a particular actor. It involves a planning algorithm, is fully automatic, and works with dependencies between agents. It suggests measures for global evaluation, but does not implement them, instead performing local evaluation in terms of plan cost. The procedure requires additional information in the form of cost and of actor capabilities, defining what dependencies in a model an actor can and cannot satisfy.

Table 24: Classification of (Bryl et al., 2006a)

	Approach							Analysis Results			Additional Notation Supported			Analysis Scope	
Approach	**Satisf Forwds**	**Satisf Backwds**	**Human Interv**	**Metrics**	**Plan-ning**	**Simu-lation**	**Model Check**	**Qual**	**Quant**	**Binary**	**Depend-encies**	**Soft-goals**	**Contrib. Links**	**Global**	**Local**
Bryl et al., 2006a	N	N	N	Y	Y	N	N	N	Y	Y	Y	N	N	M	Y

The approach by (Bryl et al., 2006a) is combined with a model checking approach focusing on security by (Bryl, Massacci, Mylopoulos, & Zannone, 2006b). This approach argues for the automatic derivation and selection of design alternatives early in the system development process, producing a secure system. The work is described in more detail in Section 2.3.5.

(Bryl, Giorgini, Mylopoulos, 2009a) : The approach introduced by Bryl et al. (2006a) is expanded by the same authors to include a systematic requirements analysis process. After a first description of the organization is modeled an Input Checker detects inconsistencies and suggests improvements. Inconsistencies include differences between actor capabilities and goal assignments, including delegations. These are calculated using a capability tree, where the satisfaction of goals in an AND/OR goal tree is calculated based on the assignment of leaf goals to actors. Missing capabilities (goals) are calculated for goals which cannot be satisfied.

Next, the planner generates a first alternative, which is assessed by an Evaluator. This tool can evaluate global qualitative or quantitative criteria, using specific measures or designer expertise. Here, a local complexity measure is calculated by summing the local complexity of leaf-level goals assigned to an actor. Similar to the treatment of cost in the author's earlier work (2006a), if the complexity of an actor's actions is greater than a threshold, the actor is a candidate to have a selection of its actions negated in the next planning iterations. Such metrics bear similarity to the use of metrics by Franch (2006).

The process continues until the Evaluator finds an acceptable plan, given its criteria. The plan produced by this method may also not be the optimal plan. The paper provides few examples of concrete evaluation metrics for the Evaluator, although this is listed as an area of future work. The overall approach is validated using the same E-Voting case study as used by (Bryl, Dalpiaz, Ferrario, Mattioli, & Villafiorita, 2009b).

Classification: We classify this work similarly to the 2006a approach. In this case, global qualitative and quantitative measures are mentioned, but specific detailed examples are not provided.

Table 25: Classification of (Bryl et al., 2009a)

	Approach							Analysis Results			Additional Notation Supported			Analysis Scope	
Approach	**Satisf Forwds**	**Satisf Backwds**	**Human Interv**	**Metrics**	**Plan-ning**	**Simu-lation**	**Model Check**	**Qual**	**Quant**	**Binary**	**Depend-encies**	**Soft-goals**	**Contrib. Links**	**Global**	**Local**
Bryl et al., 2009	N	N	Y	Y	Y	N	Y	M	Y	Y	Y	N	N	M	Y

(Asnar, Bryl, & Giorgini, 2007): Work by Asnar et al. (2007) combines the approaches of (Asnar & Giorgini, 2006), in evaluating design options in a model in terms of risk, and (Bryl et al., 2006a; 2009a), in applying a planner to find satisfactory design options. This work applies the first order logic used by (Bryl et al., 2006a; 2009a), with the addition of formalisms for criticality and goal relaxation. Here, criticality is an axiom describing the minimum level of trust between actors required for delegation of a goal, while goal relaxation describes an action where this criticality for a goal is lowered. The criticality level in delegations between actors is described in a further axiom.

This work uses the same approach as (Bryl et al., 2009a) of finding a plan, evaluating it, and then potentially replanning if the plan does not meet the desired criteria. In this work plans are evaluated in terms of risk, especially applicable to safety-critical multi-agent systems. The first type of risk considered is the same as used by (Asnar & Giorgini, 2006), the Den value of a goal, in this case qualitative. The second type of risk occurs when the criticality level of a goal is higher than the criticality level permitted between agents, occurring with goal relaxation. Their approach first tries to find a plan with no relaxation, then, failing that, tries with relaxation. The first type of risk is evaluated for a plan using the forwards propagation introduce by (Giorgini et al., 2002, 2004a, 2004b). If any top goal Den values are above a manually set threshold, backwards propagation, following (Giorgini et al., 2004b), is used to find an acceptable set of Den values for the model leaf goals. This acceptable set is used to determine which goals to "turn-off" in the plan refinement stage. Risk in terms of actor delegations is then analyzed by trying to ensure that the delegation of a goal is not relaxed by an actor who does not own the goal. Designer intervention is needed to allow exceptions to this rule via additions to a whitelist. This work uses an Air Traffic Management case study to validate their approach.

Classification: We classify this work as using both satisfaction analysis and metrics, in the form of risk analysis across actors. It uses qualitative evaluation, planning, both forwards and backwards analysis, and human intervention. The procedure is applied to agent-goal models with dependencies and takes a global view of evaluation. In Table 38 , the procedure requires the addition of actor capabilities in the form of predicates, considers trust, but does not work with cost measures.

Table 26: Classification of (Asnar et al., 2007)

	Approach							Analysis Results			Additional Notation Supported			Analysis Scope	
Approach	**Satisf Forwds**	**Satisf Backwds**	**Human Interv**	**Metrics**	**Plan-ning**	**Simu-lation**	**Model Check**	**Qual**	**Quant**	**Binary**	**Depend-encies**	**Soft-goals**	**Contrib. Links**	**Global**	**Local**
Asnar et al., (2007)	Y	Y	Y	Y	Y	N	N	Y	N	Y	Y	M	Y	Y	N

(Liaskos, McIlraith, Sohrabi, & Mylopoulos, 2011): Liaskos et al. introduce an approach to select between alternative solutions in a goal model, using quantitative preferences collected over model criteria goals. Typical goal model syntax is expanded to allow for temporal constraints (certain tasks must or must not be performed before others), optional goals, and preference goals. The approach uses a planner to find a sequence of tasks which would satisfy all mandatory goals, including the precedence constraints. Goal trees are restricted to AND/OR decomposition and Make/Break links. The approach allows for the addition of optional temporal preferences over goals, for e.g., it would be nice if one goal precedes the other. Both preferences (optional goals) and temporal preferences are given quantitative priorities to reflect their relative importance. Once a plan is found, the degree to which is satisfies preferences is calculated by summing the quantitative priorities for each preference or temporal preference satisfied.

The reasoning approach is unique in that it explicitly recommends an iterative process of preference gathering and model improvement, providing a few example improvements over sample models prompted by procedure application. Although this approach advocates model improvement and domain understanding as a consequence of analysis, it does not apply the procedure in a participatory setting, or collect any counts of model changes.

Classification: This approach users a planner to find permissible solutions to mandatory goals, then evaluates the plans by summing priorities. Human intervention is advocated to examine plans and refine the model; however the each run of the analysis procedure is automated. Results are quantitative and global over the model. The approach does not support dependencies or contributions, and does not explicitly support softgoals, but recommends what other frameworks would consider softgoals as candidates for model preferences.

Table 27: Classification of (Liaskos et al., 2011)

	Approach							Analysis Results			Additional Notation Supported			Analysis Scope	
Approach	**Satisf Forwds**	**Satisf Backwds**	**Human Interv**	**Metrics**	**Plan-ning**	**Simu-lation**	**Model Check**	**Qual**	**Quant**	**Binary**	**Depend-encies**	**Soft-goals**	**Contrib. Links**	**Global**	**Local**
Liaskos et al., 2011	N	N	M	N	Y	N	N	Y	Y	Y	N	M	N	Y	N

Discussion: Although the automatic generation of plans could be time-saving for designers, the quality of the plan found depends on the level of specification of the capabilities of each actor, requiring, in the worst case, a pair-wise comparison of each goal of each actor to each goal of each other actor. This process could be quite time consuming.

The planning approach described by Bryl et al., (2006a; 2009a) does not produce an optimal solution. The authors argue that this is acceptable, as human designers also do not aim for the optimal design, being able to identify a satisfying design, as per (Simon, 1969). Although human designers are able to balance various factors in the design process, including time and budget pressures, if human designers knew that there may be another potential design solution within the constraints of the design problem which better satisfied the problem, they would always be interested in knowing about this solution. The discovery of optimal plans given the goal model specification would be a useful capability.

The approach by (Bryl et al., 2006a) argues that the simplicity and abstraction contained in requirements models, as opposed to code, better allows for automated reasoning. However, they neglect to consider the social complexities of requirements, especially high-level requirements as represented in goal models. Such requirements are an abstraction and gross simplification of a complex socio-technical domain, with individual motivations, subtle relationships, and hidden agendas. Although making these complex relationships at least partially explicit with the aid of goal model constructs can be useful for understanding, communication and even analysis, the high level of abstraction, approximation and incompleteness makes such models inappropriate for fully automated analysis.

This line of work also does not sufficiently consider the trust or confidence of the developer in the method. In the methodology, the designer is “kept in the loop”, and is expected to refine, amend, and approve the resulting design. However, this assumes that the designer has sufficient trust in the output to accept potential solutions, even with some modification. Likely this trust comes from a deeper understanding of how those alternatives are selected which may or may not be feasible given the expertise of the designer. Furthermore, this trust relies on the developer believing that the relevant models are entirely complete and correct - something which may not be possible for socio-technical models. These procedures are interactive, but assume model accuracy and completeness, without explicitly aiming to improve model quality through iteration.

Table 28: Summary of Classifications for Planning Approaches

	Approach							**Analysis Results**			**Additional Notation Supported**			**Analysis Scope**	
Approach	**Satisf Forwds**	**Satisf Backwds**	**Human Interv**	**Metrics**	**Plan-ning**	**Simu-lation**	**Model Check**	**Qual**	**Quant**	**Binary**	**Depend-encies**	**Soft-goals**	**Contrib. Links**	**Global**	**Local**
Bryl et al., 2006a	N	N	N	Y	Y	N	N	N	Y	Y	Y	N	N	M	Y
Bryl et al., 2009a	N	N	Y	Y	Y	N	Y	M	Y	Y	Y	N	N	M	Y
Asnar et al., 2007	Y	Y	Y	Y	Y	N	N	Y	N	Y	Y	M	Y	Y	N
Liaskos et al., 2011	N	N	M	N	Y	N	N	Y	Y	Y	N	M	N	Y	N

2.3.4 Simulation

Methods have been proposed to add dynamic, temporal aspects to agent-goal models, allowing simulations of the represented network.

(Gans, Jarke, Lakemeyer, & Vits, 2002): Gans et al., (2002) have added temporal information to i* concepts, allowing for analysis using a form of simulation. This work uses an extension of i* in the SNet Framework, using task pre- and post-conditions, automatically translating i* SR models represented in Telos to ConGolog programs, allowing simulation of various scenarios and the discovery of new model properties. Their use of a ConceptBase metadata manager based on Telos to represent extended i* models allows them to perform static checks on the model. The translation into ConGolog does not include softgoals or contribution links.

The formal foundation of ConGolog is situation calculus, a variant of first-order logic, representing preconditions and the effects of actions. All terms in the representation are ordinary objects, actions, or situations. The initial situation is represented with S0 and do(a, s) means that the situation s follows after the action a. Relations, whose truth values change in situations, are called relational fluents, functions varying across situations are called functional fluents. Poss(a, s) means that a is executable in situation s. They define the domain using clauses which are primitive fluents, primitive actions and exogenous actions. The fluents or conditions in the SR model are translated into relational fluents. The ownership of resources is also represented as relational fluents, often used as pre and post conditions for an action. Primitive actions are made from the leaf tasks in the model that are not exogenous. Clauses for the simulation are derived automatically from the SR model using queries of the model stored in ConceptBase.

For each actor in the model, there is a procedure that describes its behavior. This defines which tasks occur after what conditions. Tasks in an AND relationships must execute concurrently using the conc operator. Tasks in OR relationships execute using tryAll, which executes all options concurrently and stops when one stops.

Poss(a, s) is computed for each action by looking at the preconditions in the model. During a simulation, a user can invoke exogenous actions, which are also primitive tasks, interactively. The affects of actions on fluents can be determined by looking at post conditions in the model, resetting interrupts after the action has occurred, examining changes in ownership for resource-fluents, and looking at clocktick actions for the time fluent.

Classification: This work deals with the binary satisfaction of elements, applying a simulation algorithm. It also allows for the checking of properties after model construction. As users invoke exogenous actions, a "Y" is added to the Human Intervention column. The approach supports dependencies, and a simulation occurs across actors, giving it a global scope. Information is added to Table 38 reflecting the approaches use of temporal information, conditions, and speech acts.

Table 29: Classification of (Gans et al., 2002)

	Approach							Analysis Results			Additional Notation Supported			Analysis Scope	
Approach	**Satisf Forwds**	**Satisf Backwds**	**Human Interv**	**Metrics**	**Plan-ning**	**Simu-lation**	**Model Check**	**Qual**	**Quant**	**Binary**	**Depend-encies**	**Soft-goals**	**Contrib. Links**	**Global**	**Local**
Gans et al. (2002)	N	N	Y	N	N	Y	Y	N	N	Y	Y	N	N	Y	N

(Gans, Jarke, Lakemeyer, & Schmitz, 2003a): Gans et al. (2003a) extend their previous work, allowing agents to reason over alternatives for achieving goals by comparing the quantitative effects of alternatives on softgoals. Softgoals are more fully incorporated into the procedure, given precise quantitative interpretations so they can be used as criteria in utility measures. Here, a decision-theoretic planner is used in the simulation to select the best alternative for a single goal in terms of utility functions based on that alternative's quantitative contributions to softgoals. Alternatives may include delegations to other agents, for which a process for the analysis of potential delegations is introduced. Namely, delegation is implemented with a communication protocol: the delegator sends a request to the delegate, including relevant preferences in softgoals with suitable weighting and the earliest possible starting time for the delegatee to start the job. The delegatee answers with possible softgoal satisfaction and when the job would be finished. The delegatee informs the delegator whether he is given the job.

Each deterministic program derived from the possible choices in the nondeterministic program is generated and processed to calculate the utilities and duration of the job, and delegation and negotiation is done. The utility of these programs are computed and compared to all other alternatives and the best deterministic program is returned. When the best alternative is selected, the losing delegatees are notified. The solution is globally optimal in terms of the utility functions.

Classification: This classification differs from (Gans et al., 2002) in that it now incorporates quantitative evaluation of utilities. As these utility functions are analogous to softgoal contributions, we chose to classify this as satisfaction analysis and not metrics, similar to the quantitative evaluation in methods like those introduced by Giorgini et al. (2004a). This approach also incorporates a planner. The scope of this method is difficult to classify as contributions to each softgoal in this approach are considered separately for each goal alternative, and the analysis of each decision is considered individually. Therefore, this approach does not analyze the interactions and tradeoffs between alternatives for different goals, and does not provide an overall assessment of softgoal satisfaction given a selection over the entire set of choices in the model. Analysis is potentially global across actors but local in terms of individual alternatives. We insert an "M" for these categories.

Table 30: Classification for (Gans et al., 2003a)

	Approach							Analysis Results			Additional Notation Supported			Analysis Scope	
Approach	**Satisf Forwds**	**Satisf Backwds**	**Human Interv**	**Metrics**	**Plan-ning**	**Simu-lation**	**Model Check**	**Qual**	**Quant**	**Binary**	**Depend-encies**	**Soft-goals**	**Contrib. Links**	**Global**	**Local**
Gans et al, 2003a	N	N	Y	N	Y	Y	Y	N	Y	Y	Y	Y	Y	M	M

(Gans, Schmitz, Arzdorf, Jarke, & Lakemeyer, 2005): Gans et al. expand the SNet Framework is expanded to allow for the use of roles, the monitoring of delegations, and the evolution of agents, all in a simulation of concepts derived from extended i* models (2002). This work introduces roles to cover redundant capabilities of actor instances. Many actors can do the same things, so it is better to combine these capabilities together into roles. Roles are parameterized by the duration of the tasks they perform and the contribution towards softgoals. In this way, two different roles can do the same thing, but take a different amount of time and have different effects. Delegators make choices between agents at runtime.

In order to monitor delegations, the work distinguishes between a rationale layer and an activity layer, and introduces three phases: initiating monitoring, gathering information, and drawing conclusions. The rationale layer shows how the delegator decides the monitoring importance, based on experiences with the delegate, expectation of progress, and costs for monitoring activities. Initiating monitoring uses similar criteria to decide whether or not to monitor a delegation. Gathering information is done at different levels, with monitoring information in terms of utility functions over softgoal contributions. Drawing Conclusions is done after evidence is gathered, with expectations compared to measures, and actions being taken. Delegators must have expected values for each softgoal contribution in comparison to monitored values. Results are used to determine how often monitoring occurs.

In order to capture the development of different agents over time, this approach uses meta-agents. They define a relational fluent, roles(Agent, Role, s) that allows agents to dynamically learn skills and be modified within the simulation. Instances of agents conditionally evolve to different stages within the simulation by losing and acquiring roles.

Classification: This paper expands the agent-oriented capabilities of previous work, but receives no further classifications, having the same classifications as (Gans et al., 2003a).

(Gans, Jarke, Kethers, & Lakemeyer, 2003b; Gans et al., 2001): In work similar to (Gans et al., 2005), the Trust-Confidence-Distrust (TCD) method uses notions of trust in individuals, confidence in a network, and distrust in both to help shape the dynamic formulation and simulation of i* networks in ConGolog using the SNet Framework. This approach argues that there are differences between trust in individuals and confidence in the network. Networks need to develop rules to regulate member behavior while members of the network need to monitor each other and the network. In such a situation, distrust, difference from the absence of trust, can build secretly and grow. The TCD method describes the success and failure of networks in terms of these concepts, arguing for a dynamic analysis of social relationships and viewpoints in Requirements Engineering. As well as using goal hierarchies mapped to plans using ConGolog, the Action Workflow speech-act framework is used, describing the cooperation process in loops of communicative actions.

In the conceptual framework, goals in an agent boundary are operationalized by plans, both of which generate dependencies outside of actors. Speech acts refine plans and dependencies coordinate speech acts. These relationships are linked to models for trust, confidence and distrust, which are linked to each other. Depending on the level of trust, confidence, and distrust both plans and speech act cycles are adjusted. This work does not make use of goals, subgoals, or softgoals in their models, only tasks and conditions.

Classification: In classifying this work, we chose to view the measure and simulation of trust, distrust and confidence as a form of quantitative metric over the model. Although this procedure computes quantitative simulations of trust, confidence, and distrust, it measures only the binary satisfaction of tasks. This approach uses a simulation approach to examine the dynamic behavior of the model, but, unlike the approach of (Gans et al.. 2005) does not use an explicit planning approach to select a best set of actions. It is not clear if model checking is used in this approach, although the use of SNet makes it possible. Simulation is performed globally over the entire model. We add information to Table 38 reflecting the incorporation of Trust, Distrust and Confidence.

Table 31: Classification of (Gans et al., 2003b; Gans et al., 2001)

	Approach							Analysis Results			Additional Notation Supported			Analysis Scope	
Approach	**Satisf Forwds**	**Satisf Backwds**	**Human Interv**	**Metrics**	**Plan-ning**	**Simu-lation**	**Model Check**	**Qual**	**Quant**	**Binary**	**Depend-encies**	**Soft-goals**	**Contrib. Links**	**Global**	**Local**
Gans et al., 2001, 2003b	N	N	N	Y	N	Y	M	N	Y	Y	Y	N	N	Y	N

(Wang & Lesperance, 2001): A similar approach involving the mapping of i* models to ConGolog has been developed by Wang & Lesperance (2001), using a mapping which differs from the approach of Gans et al. Here, i* models are used to model alternatives, actors, actor goals and dependencies, while ConGolog is used to provide more detailed information about the system. The ConGolog models are then used to validate the system through simulation. ConGolog is able to accommodate incomplete models, either by not completely stating the initial state of the system or by including non-deterministic choices. A process simulation for ConGolog had been developed, using Prolog. At the time the paper was published, a verification tool for ConGolog was also under development.

In order to add the information to i* models needed to transform them into detailed ConGolog specifications, this approach creates annotated SR diagrams (ASR) using composition and link annotations. Composition annotations consist of sequence, alternative, concurrency, and prioritized concurrency. Link annotations indicate conditions for the performance of the subtask, and the number of times it should be performed. In mapping i* to ConGolog, the work considers only agents, positions, roles, goals, tasks, means-ends links, and decompositions, ignoring softgoals, resources, dependencies, and contributions links. To deal with the presence of dependencies, they recommend that dependencies are decomposed to the necessary detailed tasks within each actor.

The work suggests a method for applying the ConGolog simulation: Step 1: Build SD Models, Step 2: Build SR models, Step 3: Build the ASR models: Step 4: Develop the initial ConGolog model, Step 5: Validate the ConGolog model by simulation and verifications, Iterate Step 1 to 5: Refining the i* and ConGolog models until objectives are met, and Step 6: Produce a requirements specification. After the simulation has been performed, the work recommends iteration over the i* and ConGolog models when issues with the simulation are found, leading to improvements in the specification. The method is demonstrated through a meeting scheduler example.

Classification: This work applies simulation to extended i* models. It considers agents and dependencies in the simulation, which is performed globally for the model. The simulation requires additional information to operate, namely composition and link annotations.

Table 32: Classification of (Wang & Lesperance, 2001)

	Approach							Analysis Results			Additional Notation Supported			Analysis Scope	
Approach	**Satisf Forwds**	**Satisf Backwds**	**Human Interv**	**Metrics**	**Plan-ning**	**Simu-lation**	**Model Check**	**Qual**	**Quant**	**Binary**	**Depend-encies**	**Soft-goals**	**Contrib. Links**	**Global**	**Local**
Wang & Lesperance, 2001	N	N	N	N	N	Y	N	N	N	Y	N	N	N	Y	N

Discussion: Model simulation can be useful to test the accuracy of and completeness of the model, and to learn new things about the domain; however, simulation requires the addition of much additional, specific information to the model, making the application of such methods laborious and more suitable for later stages of development, when such information is available. Furthermore, it is not always clear what sort of information one would derive from a simulation of goal- and agent-oriented models. What sort of qualities, effects or events should be observed in the models? Is this phenomena always domain specific or can it be generalized? What sort of actions should be taken based on the observations? Should the model be modified? Can the results lead you towards these modifications? The papers in this section focus more on the technical details of application, and less so on how to process and use the results. A summary of the classification for simulation procedures is included in Table 33.

Table 33: Summary of Classifications for Simulation Procedures

	Approach							Analysis Results			Additional Notation Supported			Analysis Scope	
Approach	**Satisf Forwds**	**Satisf Backwds**	**Human Interv**	**Metrics**	**Plan-ning**	**Simu-lation**	**Model Check**	**Qual**	**Quant**	**Binary**	**Depend-encies**	**Soft-goals**	**Contrib. Links**	**Global**	**Local**
Gans et al. 2002	N	N	Y	N	N	Y	Y	N	N	Y	Y	N	N	Y	N
Gans et al, 2003a, 2005	N	N	Y	N	Y	Y	Y	N	Y	Y	Y	Y	Y	M	M
Gans et al., 2001, 2003b	N	N	N	Y	N	Y	M	N	Y	Y	Y	N	N	Y	N

2.3.5 Model Checking

The approaches in this section add formalizations to goal models, and then use the content of the goal models to check formally expressed properties over the domain. The focus is on checking the consistency and correctness of the goals and requirements captured in the model.

(Fuxman, Pistore, Mylopoulos, & Traverso, 2001): Fuxman et al. (2001) converted i* models to Formal Tropos, a language which incorporates i* concepts with, among other things, formal expressions of creation, fulfillment, and invariant properties for goals, dependencies or actors. First order linear-time temporal logic statements are used to represent desired and required constraints over the system. The formal domain representations and constraints are converted into an intermediate language which is used as input for a symbolic model checker. In this way, formal properties are validated and consistency checks can be made. We classify this work along with the following paper.

(Fuxman, Liu, Pistore, Roveri, & Mylopoulos, 2003): The work of Fuxman et al. (2001) is extended, introducing a methodology and addressing scalability. This work adds extra syntax to i* SR models: prior-to links, showing temporal order, and cardinality constraints on goal relationships. When converting i* models to formal Tropos, actors and intentional elements are mapped to Formal Tropos classes, entities in the domain are added, modes (achieve, maintain), are added to intentions, fulfillment and creations constraints are added from some classes and entities, and these conditions are classified as sufficient (trigger), necessary (condition), or necessary and sufficient (definition).

A significant contribution of this work is the introduction of rules which allow the partial automatic translation of i* to Formal Tropos. Recognizing that the conversion into a formal specification takes effort, this work introduces rules relating the creation conditions of subgoals to the existence and fulfillment of parent goals, the fulfillment of parent goals to the fulfillment of subgoals, the entity and owners of sub and parent goals, and constraints to prior-to relationships.

After a formal representation of the model has been created, the next step is to define properties over the formal model. Here they define assertion properties, which must be true in all cases, and possibility properties, that should hold for at least one case.

They introduce the T-Tool, using the NuSMV symbolic model checker, accepting a Formal Tropos model, properties to check, and an upper bound for the class instances. The tool builds a finite model representing all behaviors of the domain and checks the properties over the model. In addition to desired possibility and assertion properties, the tool also checks for general model consistency, i.e., the model does not contradict itself and there exists at least one valid scenario respecting all constraints while instantiating all classes. Because the checks are bounded in order to deal with the state explosion problem, correctness of assertions can only be completely checked in some cases.

Overall, the results of the checker help to find errors in the specification that leads to an improved model and to a better understanding of the domain. Experiments are performed to show that the approach is practically applicable.

Classification: The approaches introduced by (Fuxman et al., 2001) and (Fuxman et al., 2003), assess the binary satisfaction of intentional elements when checking for consistency. The checks themselves run automatically, but an iterative process of manually defining the bounds of the model checker is often required. The approach supports dependencies and focuses on global analysis, although local properties could easily be defined. (Fuxman et al., 2003) introduce temporal information, conditional information, and constraints, as represented in Table 38.

Table 34: Classification for (Fuxman et al., 2001) and (Fuxman et al., 2003)

	Approach							**Analysis Results**			**Additional Notation Supported**			**Analysis Scope**	
Approach	**Satisf Forwds**	**Satisf Backwds**	**Human Interv**	**Metrics**	**Plan-ning**	**Simu-lation**	**Model Check**	**Qual**	**Quant**	**Binary**	**Depend-encies**	**Soft-goals**	**Contrib. Links**	**Global**	**Local**
Fuxman et al., 2001, 2003	N	N	M	N	N	N	Y	N	N	Y	Y	Y	N	Y	M

(Giorgini, Massacci, Mylopoulos, & Zannone, 2004c): Work by (Giorgini et al., 2004c) argues that security and trust issues are not currently considered early enough in system development. To this end, they adjust i*/Tropos to take trust into account. Analysis in this method focuses on extensions which separate trust dependencies from functional dependencies, distinguishes ownership and considers the delegation of permissions. They represent these ideas using formal predicates and check their models using datalog, accepting a logic program composed of a set of rules representing the model. Checks are performed for consistency, making sure there are no contradictions, then the trust and delegation of the model is checked for correctness. The formal approach in this method does not explicitly consider non-functional requirements or softgoals.

Classification: This method considers the global, binary satisfaction of functional requirements. It uses a model checking algorithmic approach and considers dependencies, but not softgoals or contribution links. It also considers delegation and ownership, as expressed in Table 38.

Table 35: Classification for (Giorgini et al., 2004c)

	Approach							Analysis Results			Additional Notation Supported			Analysis Scope	
Approach	**Satisf Forwds**	**Satisf Backwds**	**Human Interv**	**Metrics**	**Plan-ning**	**Simu-lation**	**Model Check**	**Qual**	**Quant**	**Binary**	**Depend-encies**	**Soft-goals**	**Contrib. Links**	**Global**	**Local**
Giorgini et al., 2004c	N	N	N	N	N	N	Y	N	N	Y	Y	N	N	Y	N

(Bryl, Massacci, Mylopoulos, & Zannone, 2006b): Work by (Bryl et al., 2006b) combines the work of (Giorgini et al., 2004c) with the automatic derivation and analysis of design alternatives in (Bryl et al., 2006a). As mentioned, this approach argues for the automatic derivation and selection of design alternatives early in the system development process, producing a secure system. As with other work along this line, the resulting design is sufficient, but not necessarily optimal.

As with (Giorgini et al., 2004c) , the suggested process consists of modeling the system, translating the model automatically into clauses, and verifying design or security properties in the model. To facilitate the planning aspects, domain actors, desires, capabilities, entitlements, goal decompositions and trust relationships (execution or permission) are identified.

The process of gaining trust is encoded using axioms for negotiation, contract, delegations under suspicion, and evaluation. Because of the concern for privacy and security, the planner finds only need-to-know plans, where only necessary actors participate.

The approach is verified via a case study of a Medical Information System. This approach does not explicitly consider NFRs or partial satisfaction.

Classification: We classify this approach as finding the satisfaction of goals in a binary manner. It uses both a planning and a model checking approach and supports agent-oriented analysis with global analysis. It adds information concerning actor capabilities, ownership, delegation, and trust.

Table 36: Classification for (Bryl et al., 2006b)

	Approach							Analysis Results			Additional Notation Supported			Analysis Scope	
Approach	**Satisf Forwds**	**Satisf Backwds**	**Human Interv**	**Metrics**	**Plan-ning**	**Simu-lation**	**Model Check**	**Qual**	**Quant**	**Binary**	**Depend-encies**	**Soft-goals**	**Contrib. Links**	**Global**	**Local**
Bryl et al., 2006b	N	N	N	N	Y	N	Y	N	N	Y	Y	N	N	Y	N

Discussion: Although these approaches show promise for identifying issues in models, including inconsistent and missing requirements, detailed system information, especially concerning temporal properties, is required, making this approach appropriate for later stages of system analysis. Furthermore, it may not always be clear what properties one would like to check over the model, especially if the extra information added to the model is missing. In considering the application of model checking to regular goal- and agent-oriented models, not enhanced with extra information, intuitive and meaningful properties over the model may or may not exist. More investigation into this area is needed. A summary of model checking techniques is provided in Table 37.

Table 37: Summary of the Classifications for Model Checking Procedures

	Approach							Analysis Results			Additional Notation Supported			Analysis Scope	
Approach	**Satisf Forwds**	**Satisf Backwds**	**Human Interv**	**Metrics**	**Plan-ning**	**Simu-lation**	**Model Check**	**Qual**	**Quant**	**Binary**	**Depend-encies**	**Soft-goals**	**Contrib. Links**	**Global**	**Local**
Fuxman et al., 2001, 2003	N	N	M	N	N	N	Y	N	N	Y	Y	Y	N	Y	M
Giorgini et al., 2004c	N	N	N	N	N	N	Y	N	N	Y	Y	N	N	Y	N
Bryl et al., 2006b	N	N	N	N	Y	N	Y	N	N	Y	Y	N	N	Y	N

2.3.6 External Consistency Checking

Several approaches have developed the practice of checking goal- and agent-oriented models against other types of models representing the same domain, for consistency checks. Although each type of model may capture a different range of information, overlaps between this information exists, and checks can be defined over these overlaps to increase the completeness of each type of model. Although these types of methods can be considered as a type of analysis, this analysis requires the presence of other types of models. Therefore it is difficult to classify it along with the other methods in our summary tables.

(Maiden, Jones, Manning, Greenwood, & Renou, 2004): Work in (Maiden et al., 2004) defines a Requirement Analysis approach (RESCUE) involving several different model artifacts, include i* models. This method produces use case models with scenario descriptions, i* models, and human activity models of the same domain. A fourth "stream", along with the three types of models, is used to manage requirements derived from each of the model types. They use five synchronization stages to compare models, often as part of a workshop with stakeholders. They have created mapping between concepts contained within the three types of models to facilitate checks between models. For example, a task in i* maps to an activity in a use case and human activity model, a goal maps to a requirement in a use case model and a goal in a human activity model, etc. Each synchronization stage contains a series of checks to perform across models. The work describes the checks of the first two stages: determining system boundaries and determining work allocation (assigning tasks). In a detailed case study involving an air traffic control system, these checks were able to produce a number of issues and missing concepts within each model. Addressing these issues improved the quality of the model and the resulting specification.

Work in (Grau et al., 2005b) and (Grau et al., 2008) adopts the rescue approach in its initial stages, checking different model artifacts (human activity models, detailed interaction scripts, and i* models) for consistency with each other. More details on this work can be found in Section 2.4.

(Gordijn, Petit, & Wieringa, 2006): This work focuses on collectives of organizations which band together to create networks which satisfy the needs of customers and provide financial value to the network participants. They use goal-oriented RE to represent these networks, focusing on sustainability instead of satisfiability. They focus on the view point of the business manager, who requires a model of business goals and a value model. In modeling goals they us a subset of i* syntax, adding constructs for property, a variable, scale, a set of values associated with a property, value, an element in the scale, state, the association of a value to a property, and causal relationships between properties. It is not clear how these constructs are actually used in their example i* models. The work also develops value models, which model actors, value objects, value ports, value interfaces, value exchange, market segments, value activities, and dependency paths. Given a value model with real number attributes, Net Value Sheets, showing net cash flows, and Discounted Net Present Value, showing evolution over time, can be calculated. However, in this work, the numeric values are not collected and these value calculations are not computed.

The approach does not explicitly say how the two types of models, i* and value, are linked, only explaining specific relationships between the models used in their example.

Generally, this approach takes advantage of the different views offered by each type of model to lead to improvements in each model and a general improved understanding of the domain. Goals provide the intentional rationale for value models, while value models help to point out missing aspects in goal models. This work does not apply any systematic or algorithmic analysis of goal or value models.

(Stirna & Persson, 2007): The EKD (Enterprise Knowledge Development) Modeling process described by Stirna & Persson (2007) uses goal modeling along with other modeling frameworks as part of enterprise modeling. In this process, goal models are created along with five other types of sub-models, questions are used to drive the creation of inter-model links. Participatory, “consensus-driven” modeling is favored over “consultative” participation.

Discussion: The comparison of a goal model to another type of model from the domain in order to find missing elements can be highly beneficial. Of course, effort is required to build the other models, but if a process that already builds other models is employed, or the domain is such that other types of models help to capture areas missed by goal models (dynamic behavior, for example), then comparison can be quite beneficial. An efficient comparison between different types of models is aided by ideas concerning how concepts in one model map to concepts in another. These ideas manifest themselves in checklists or rules in the above approaches. Such rules are likely necessary in order for an effective comparison.

2.3.7 Classification Summary

The combined version of the summary tables is presented in Table 38. The information required by each procedure is summarized in Table 38.

Table 38: Combined Classification for Goal Model Analysis Procedures

	Approach							Analysis Results			Additional Notation Supported			Analysis Scope	
Approach	**Satisf Forwds**	**Satisf Backwds**	**Human Interv**	**Metrics**	**Plan-ning**	**Simu-lation**	**Model Check**	**Qual**	**Quant**	**Binary**	**Depend-encies**	**Soft-goals**	**Contrib. Links**	**Global**	**Local**
Maiden et al., 2007	Y	N	Y	N	N	N	N	N	N	Y	Y	Y	Y	Y	N
Jureta et al., 2008, 2010	N	Y	N	N	N	N	N	N	N	Y	N	Y	N	Y	N
Chung et al., 2000	Y	N	Y	N	N	N	N	Y	N	Y	N	Y	Y	Y	N
Giorgini et al., 2002, 2004a	Y	N	N	N	N	N	N	Y	Y	Y	N	M	Y	Y	N
Giorgini et al., 2004b	Y	Y	N	N	N	N	N	Y	N	Y	N	M	Y	Y	N
Giorgini et al., 2005	Y	Y	N	N	N	N	N	Y	N	Y	M	Y	Y	M	Y
Ernst et al., 2010	Y	Y	M	N	N	N	N	Y	N	Y	N	M	Y	Y	Y
i* Evaluation, Horkoff, 2006	Y	N	Y	N	N	N	N	Y	N	Y	Y	Y	Y	Y	N
Z.151, 2008, Amyot et al., 2010, Pourshahid et al., 2011	Y	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y	Y
Asnar & Giorgini, 2006	Y	Y	N	N	N	N	N	Y	M	Y	N	M	Y	Y	N
Barone et al., 2011	Y	N	N	N	N	N	N	Y	Y	Y	N	M	Y	Y	N
Letier & van Lamsweerde, 2004	Y	Y	N	N	N	N	N	N	Y	Y	M	N	N	Y	N
van Lamsweerde, 2009	Y	N	N	N	N	N	N	N	Y	Y	N	Y	M	Y	N
Franch & Maiden, 2003	N	N	N	Y	N	N	N	N	Y	N	Y	Y	N	Y	N
Franch et al., 2004	N	N	N	Y	N	N	N	M	Y	N	Y	Y	N	Y	Y
Franch, 2006	N	N	Y	Y	N	N	N	Y	Y	N	Y	Y	Y	Y	Y
Kaiya et al, 2002	N	N	N	Y	N	N	N	N	Y	Y	N	N	M	Y	N
Tanabe et al., 2008	Y	N	N	Y	N	N	N	N	Y	Y	N	N	M	Y	N
Bryl et al., 2006a	N	N	N	Y	Y	N	N	N	Y	Y	Y	N	N	M	Y
Bryl et al., 2009a	N	N	Y	Y	Y	N	Y	M	Y	Y	Y	N	N	M	Y
Asnar et al., 2007	Y	Y	Y	Y	Y	N	N	Y	N	Y	Y	M	Y	Y	N
Liaskos et al., 2011	N	N	M	N	Y	N	N	Y	Y	Y	N	M	N	Y	N
Gans et al. 2002	N	N	Y	N	N	Y	Y	N	N	Y	Y	N	N	Y	N
Gans et al, 2003a, 2004	N	N	Y	N	Y	Y	Y	N	Y	Y	Y	Y	Y	M	M
Gans et al., 2001, 2003b	N	N	N	Y	N	Y	M	N	Y	Y	Y	N	N	Y	N
Wang & Lesperance, 2001	N	N	N	N	N	Y	N	N	N	Y	N	N	N	Y	N
Fuxman et al., 2001, 2003	N	N	M	N	N	N	Y	N	N	Y	Y	Y	N	Y	M
Giorgini et al., 2004c	N	N	N	N	N	N	Y	N	N	Y	Y	N	N	Y	N
Bryl et al., 2006b	N	N	N	N	Y	N	Y	N	N	Y	Y	N	N	Y	N

Table 39: Information Required by Each Procedure

	Additional Information	Required by
1	Goal Cost	Satisfaction Analysis: (Giorgini et al., 2005)(Asnar et al., 2007) (Giorgini et al., 2004b)(Asnar & Giorgini, 2006), Planning: (Bryl et al., 2006a)
2	Risk	Satisfaction Analysis: (Asnar & Giorgini, 2006), Planning: (Asnar et al., 2007)
3	Textual Arguments	Satisfaction Analysis:(Maiden et al., 2007) , Metrics, Model Checking: (Kaiya et al., 2002)
4	Probabilistic Information	Satisfaction Analysis: (Giorgini et al., 2005) (Letier & Lamsweerde, 2004)
5	Events and Treatments	Satisfaction Analysis: (Asnar & Giorgini, 2006)
6	Importance/Priority	Satisfaction Analysis: (Asnar & Giorgini, 2006) Planning: (Liaskos et al., 2011)
7	Actor Capabilities	Planning: (Bryl et al., 2006a, 2007) (Asnar et al., 2007), Model Checking: (Bryl et al., 2006a):
8	(Pre/Post) Conditions/ Temporal Information	Planning: (Liaskos et al., 2011) Simulation: (X. Wang & Lespérance, 2001) (Gans et al., 2003a) (Gans et al., 2005) (Gans et al., 2003b), Model Checking: (Fuxman et al., 2001) (Fuxman et al., 2003)
9	Delegation/Ownership	Model Checking: (Gans et al., 2002) (Bryl et al., 2006b):
10	Trust	Planning: (Asnar et al., 2007), Simulation: (Gans et al., 2003b), Model Checking: (Giorgini et al., 2004c) (Bryl et al., 2006b):
11	Speech Acts	Simulation: (Gans et al., 2003b)
12	Confidence and Distrust	Simulation: (Gans et al., 2003b)
13	Preferences	Satisfaction Analysis: (Jureta et al., 2008, 2010), (Ernst et al. 2010) Planning: (Liaskos et al., 2011) Model Checking: (Kaiya et al., 2002)
14	Cardinalities	Simulation:(X. Wang & Lespérance, 2001), Model Checking: (Fuxman et al., 2003)
15	Domain specific formula	Satisfaction Analysis: (A Pourshahid et al., 2008) (Barone et al., 2011) (Letier & Lamsweerde, 2004)
16	KPIs/Metrics/Gauges	Satisfaction Analysis: (Pourshahid et al., 2008) (Pourshahid et al., 2011) (Barone et al., 2011) (Lamsweerde, 2009)
17	Mandatory/Optional Requirements	Satisfaction Analysis: (Jureta et al., 2008, 2010), (Ernst et al., 2010) Planning: (Liaskos et al., 2011)

2.4 Model Development Approaches

Several methods have been introduced which focus on guiding modelers through the process of model construction, often for some specific purpose such as business process reengineering or deriving alternative architectures. In addition to guiding users, these procedures aim to increase the accuracy and completeness of models, addressing some of the identified challenges of early RE analysis. We include a brief review of these methods in the following section.

(Letier & van Lamsweerde, 2002): Work in (Letier & van Lamsweerde, 2002) introduces tactics for refining and elaborating goal models in the KAOS Framework. This approach does not specifically advocate improving the quality of the model, but by guiding the user in model refinement, the model becomes more complete and likely more stylistically consistent with other models refined with the same tactics. The work first formally defines what it means for a goal to be realizable in terms of the constructs within the KAOS Framework, basically meaning that an agent has control over all of the monitored and controlled variables necessary to accomplish a goal and that an agent is able to restrict its behavior to ensure the goal. Next, they formally check whether goals are realizable by existing agents. A technique is described to find new agents and their capabilities, and to refine goals into sub-goals which are realizable by agents. They use specification elaboration tactics, each having an associated formal pattern, to guide elaboration based on the unrealizability conditions found in the previous steps. The entire procedure is illustrated with the London Ambulance System Example.

(Grau, Franch, Mayol, Ayala, & Cares, 2005a): (Grau et al., 2005a) introduces the RiSD method to guide the systematic construction of i* strategic dependency models. This method aims to reduce the uncertainty in model construction, reduce the size of the model and add traceability to the models. The first phase in this methodology focuses on constructing the social system model iteratively, identifying an initial set of system actors and their main goals, identifying dependencies, determining types using focused questions, then analyzing the model to look for missing components. Next, the software system, its main goals, subsystems and resulting dependencies are iteratively added to the model. In creating i* elements, they apply structured guidelines for element names, depending on the element type. To support traceability, a new "supports" link is added to the SD models, to show actors which support one another, as well a new "refine" link, to show that one dependency refines another. However, these types of relationships could be derived implicitly from the corresponding SR models.

(Grau, Franch, & Maiden, 2005b): In (Grau et al., 2005b), the argument is made that business process reengineering and system development are one in the same and that, although the i* language can be useful for business process reengineering, lack of methodologies for systematically creating i* models makes them inconsistent and unreliable for finding design alternatives. To this end, the work describes a detailed method for creating i* and other models of the system, including systematic ways to derive "to-be" model alternatives.

The methodology has five steps: analyze the current process, building i* models of the current process, reengineering the system, evaluating alternatives and defining the new system. They focus in this paper on the second and third step, referring to existing work for the other steps, including the evaluation of model alternatives. The first step adopts the RESCUE approach (Maiden et al., 2004), developing context models, Data-Flow-Diagrams and Human Activity Models of the current system. In stage two, the "as-is" model is built systematically, separated into two parts: the functional, descriptive part and the strategic, intentional part. First, actors are identified and modeled, then the operational i* model is built. In building the i* model, the artifacts developed in step one are used, along with Detailed Interaction Scripts (DIS), which capture scenarios. Rules are used to convert DIS descriptions to i* constructs. The intentional model is built next by asking questions about activities (Why? For whom? By whom?). A quality attribute category is used to generate questions automatically, helping to discover goals and softgoals. Consistency between model artifacts is checked, as described in Section 2.3.6.

In stage three, patterns from the KAOS approach (Dardenne et al., 1993) are used to find goals for the new system, with existing goals classified as maintain or avoid. Optimize goals are added to the model and questions are again used to find additional goals. New actors are found and responsibilities are reallocated using patterns in order to make model changes uniform. The final stage of step 3 involves checking for consistency between the original i* model and alternatives using a checklist of consistency checks.

(Grau & Franch, 2007): The work of (Grau et al., 2005a) is continued in (Grau & Franch, 2007), combining the systematic generation of alternative architectures with metrics to evaluate the architectures. The aim of the work is to provide early support for the exploration and evaluation of alternative architectures, using goal models. Again, they introduce a complex methodology, here part of a reengineering framework call ReeF, refined in this work to SARiM (Software Architecture Reengineering i* Method), the steps of which are similar to the steps in (Grau et al., 2005b), but with a focus on system architectures.

This approach focuses on SD models, using existing architectural patterns to derive model alternatives. Patterns are selected by comparing quality attributes. Once a pattern is selected it is connected to an i* model by mapping architectural dimensions to i* concepts. The "base" i* model and the pattern model are compared to match elements and actors, and the models are selectively merged, adding needed dependencies.

The satisfaction of alternative SD models is evaluated using metrics, as described in Franch et al. (2004). The method is demonstrated using a case study involving the Home Service Robot.

(Grau, Franch, & Maiden, 2008): The approach introduced in (Grau et al., 2005b) is expanded and described in more detail in (Grau et al., 2008), with the entire process given the name of PRiM (Process Reengineering i* Method). Comparing (Grau et al., 2005b) to (Grau et al., 2008), the methodology has gained an extra step, reengineering the current process, being promoted from step 3.1 in (Grau et al., 2005b) to phase 4 in (Grau et al., 2008). Additional step refinement is added to phase/step 5, evaluation of alternatives. In phase/step 2, building the i* model of the current process, the rules to convert DIS models to i* are modified, adding rules for modeling reflexive actions and alternative courses of actions. Phase/step 2.3, building of the intentional model, now contains guidelines to drive model creation. Step/phase 4.2, the reallocating responsibilities step of the generation of alternatives, has acquired two new patterns to direct the allocation of responsibilities. These patterns now consist of goal achievement, goal delegation, goal operationalization, and softgoal operationalization.

Much more detail is provided for step/phase 4.3, checking for consistency amongst alternatives. Here they check for intentional equivalence, where two models have the same goals and operationalization of the same dependee goals, and intentional inclusion, where one model has at least the intensions of another.

In step/phase 5, the evaluation of alternatives, the REACT method, as described by (Franch & Maiden, 2003) and (Franch et al., 2004) is used, evaluating properties using structural metrics. Finally, the work describes tool support in the form of REDEPEND-REACT and J-PRiM.

Discussion: The work which introduces structured methodologies for the creation of goal models introduces several positive ideas, but also has its drawbacks. In a positive light, methods to support the creation of goal, or i* models, would help new users and make resulting models more consistent. The use of quality questions in (Grau et al., 2005b) to find goals could be helpful, especially for new modelers, and conversion from DIS descriptions to i* could be a useful way to start a model.

On the downside, one of the benefits of goal modeling is its flexibility and expressiveness, the ability to capture the qualitative and social aspects of the domain that many other methods neglect. Having a systematic way to create such models removes some of the flexibility and creativity from the modeling process, making it more difficult to capture ideas as they arrive. If models are not intended to be the subject of automatic analysis, the strong emphasis on consistent styles may not be necessary.

Examining the approach in (Grau et al., 2005b) , the separation of the intentional and functional models could miss important interactions between the two views. In addition, the entire process in this work and (Grau et al., 2008) is quite complex, and could be difficult to understand and apply without guidance.

2.5 Related Analysis Methods

In this Chapter, thus far we have focused on approaches directly related to goal- and agent-oriented modeling. In this section, we broaden our focus and examine more general approaches to representing and reasoning over system alternatives and domain information not represented in goal models.

These approaches can serve a similar purpose as goal model analysis, including domain understanding, communication, and increased stakeholder involvement. We review a selection of methods in the field of Requirements Engineering which examine trade-off analysis and manage requirement conflicts. We turn to work in Business and look briefly at decision support systems, Balanced Scorecards and Strategy Maps, and the Business Motivation model. Finally, alternative ways to perform reasoning are considered, including qualitative reasoning in AI and multi-valued logic.

2.5.1 Alternative and Trade-off Analysis in Requirements Engineering

Alternative methods which facilitate the decision process in software design exist.

(Anton & Potts, 1998): The approach introduced in (Anton & Potts, 1998) supports the consideration of alternatives in the design process, providing a framework to represent issues, alternatives, artifacts and justifications. This approach differs from goal modeling techniques in that it focuses on modeling the design process instead of the domain, with stakeholder goals represented implicitly in the justifications. The effect and side-effects of alternatives are not represented visually, and the approach does not necessarily encourage consideration over a space of interacting alternatives. Generally, the focus is on recording design decisions and not on supporting the user in making these decisions. Despite these points, the use of structured text as a means of recording domain assumptions in the selection of alternatives can be useful.

(Feather & Cornford, 2003): Work in (Feather & Cornford, 2003) introduces a framework for detecting defects in requirements, aimed at supporting trade-off analysis early in the development process. The approach relates requirements, failure modes for requirements, and PACTs, various interventions for failure modes, together in a quantitative system of effects and impacts. The overall result is a measurement of the likelihood of success of each requirement, taking into account the importance of each requirement, as well as a measure of the costs of interventions and remaining failures. The method incorporates a notion of tradeoffs as PACTs are permitted to impact failure modes both negatively and positively. As quantitative information used in this method is based on expert judgment, as with quantitative goal model analysis, the level of approximation increases with each calculation or propagation. Although this approach offers a method to select between sets of PACTs, it requires detailed knowledge of the requirements of the system and is applicable to a later stage of system development than the goal model analysis introduced in this work.

2.5.2 Requirement Conflict and Inconsistency Management

The detection and management of conflicts among requirements or goals can be considered a sub-problem within the general field of requirements analysis. Several approaches have addressed this problem, we summarize two prominent examples in this section.

(van Lamsweerde, Darimont, & Letier, 1998): Work uses the KAOS framework as part of an effort to manage conflicts in RE. This work argues that all conflicts between goals/requirements should be detected and managed as part of system development. The authors identify several types of model inconsistencies, including for example, a process-level deviation, where an RE process rule and a process state are inconsistent, a terminology clash where a concept is given multiple syntactic names, a conflict, where assertions are logically inconsistent, and a divergence, where there is a boundary condition which makes assertions logically inconsistent. The approach suggests techniques for detecting inconsistencies and then resolving inconsistencies. Example resolution techniques include avoiding boundary conditions, weakening a goal, and finding alternative goal refinements.

The classification and resolution techniques described in (van Lamsweerde, Darimont, & Letier, 1998) are potentially useful in deciding how best to manage inconsistencies in a real world project. However, many classifications and resolution strategies are specific to concepts expressed in KAOS, e.g., process vs. instance, boundary conditions, and temporal orderings. These concepts are better suited to later RE, where this information can be more easily elicited, and where the labor required to express the RE space formally can be focused on areas of the system which are particularly important.

(Robinson, Pawlowski, & Volkov, 2003): Robinson et al. introduce the Requirements Interaction Management (RIM) Framework, reviewing existing work and introduce new techniques to formally describe, detect, classify, and manage interactions between requirements. Examples are given using the KAOS language. They classify interactions between requirements in several categories, including conflicts (negative interactions), positive interactions, and implementation conflicts. The work provides an in depth overview of many papers relating to a variety of topics in requirements interaction management. This includes approaches which describe requirements interactions using different scales (binary, qualitative, quantitative, and fuzzy logic); disciplines which address interaction management (software engineering, database integration, knowledge representation, artificial intelligence, negotiation support, social confliction and negotiation, and individual decision making); approaches for indentifying interactions (AI-based, RE-based) or detecting conflicts (classification-based, pattern-based, AI planning, scenario analysis, formal methods, and run-time monitoring); and approaches for generating a conflict resolution (relaxation, refinement, compromise, or restructuring).

The authors discuss methodological questions, including when interaction management steps should occur in a RE process. Finally, the paper provides descriptions of several projects which illustrate aspects of RIM, including Win Win, the NFR Framework, ViewPoints, KAOS, Software Cost Reduction, Deficiency-driven Requirements Analysis, and M-Telos.

Overall, the paper paints a broad and detailed picture of approaches which fall under the umbrella of RIM. However, as its approach favors the use of formal representations, we argue that many aspects of the RIM Framework are not easily applicable to very early RE. See Section 2.3.1.3 for our discussion of requirements analysis using the KAOS representation. Similar to our previous arguments, formal inconsistency detection and management could be more successfully applied in later RE stages when the focus of the project has become clear and the space of implementation alternatives has been narrowed.

2.5.3 Business Approaches

Several approaches in Business Research have attempted to aid users in making decisions over alternatives.

2.5.3.1Decision Support Systems

Decision Support Systems have evolved as a means to support management when making key business decisions (Power, 2007). These systems differ primarily from most goal model analysis procedures as they aim to support decisions over pre-existing systems based on analysis over an often large quantity of information resources. In contrast, goal- and agent-oriented models typically aim to make design decisions concerning new or redesigned systems using high-level information concerning intentions and interactions over a socio-technical network.

2.5.3.2 Balanced Scorecard & Strategy Maps

The Balanced Scorecard method, describe in (Kaplan & Norton, 1992) offers a way to present and balance between financial and operational measures for a business management audience. The scorecard looks at the business from four perspectives: appearance to customers, appearance to shareholders, what to excel at (internal perspective), and innovation and learning. By presenting these views together, managers avoid focusing only on a single area, and can see trade-offs amongst their decisions. In each area, a company must articulate specific goals, such as time to market, performance, productivity, quality, and cost, and then find specific concrete measures for each goal. They do not specify how measures in each category link together, but encourage companies to investigate these links on their own through simulation and cost modeling.

This method is comparable to goal modeling approaches reviewed earlier in that they both encourage the consideration of trade-offs among goals in decision making. However this method differs from these approaches, not only in its means of representation, but also its purpose. Most applications of goal modeling are meant for system development or process reengineering, although a few exceptions exist (Horkoff, E. Yu, & Liu, 2006). The Balance Scorecard method is instead meant for a continuous analysis of the health and prosperity of a business, involving a dynamic analysis over continually changing measures. This approach does not offer a way to perform systematic analysis beyond the calculation of specific measures, not describing methods which may allow these measures to be systematically compared or combined to produce overall results.

The Balanced Scorecard method has evolved to include the use of Strategy Maps (Kaplan & Norton, 2001). Application of the Balanced Scorecard revealed that organizations had difficulty in articulating the strategies which drove the identification of their goals and subsequent measures. Strategy Maps contain similarities to goal models, identifying high-level needs and showing how these needs can be accomplished by increasingly more specific objectives. Levels in the maps are perspectives, the same perspectives introduced with the Balanced Scorecard Method. The perspectives are ordered as follows: Financial Perspective, Customer Perspective, Internal Perspective, and Learning and Growth Perspective. Although some of the individual elements within the strategy are concretely defined, for example, Environmental: Number of incidents reduced by 63%, it is not clear how these measures effect each other, or exactly what the semantics of the links between strategy components are. The relationship is described as “X arises from Y” or “X will be achieved by Y” without stating exactly how. Essentially, there is no explicit propagation of measures throughout the graph. Considering this approach in terms of goal model evaluation does not provide ideas for methods of propagation; however, this approach, as well as the Balanced Scorecard Approach could be looked at as a source of example metrics, looking at the types of metrics companies choose to measure their performance. Of course, such measures may change depending on the domain, but metrics suggested in these approaches could be used as useful starting points in designing measurements.

2.5.3.3 The Business Motivation Model

The Business Motivation Model (*The Business Motivation Model Business Governance in a Volatile World*, 2005) was introduced as a way to represent business plans, including motivations for plans, the plans themselves, things which influence the plans and the inter and intra-relationships between plan elements, motivations, and influencers. This approach bears similarities to goal-oriented approaches in that it emphasizes the “why”, claiming that the motivations for business activities should always be identifiable. The model consists of several concepts. Desired results are described as an “End” which contains a Vision, amplified by goals, which are quantified by objectives. Thus, the goals of the organization are concretely defined. Along with Ends, the approach defines "Means" which consists of Missions which makes a Vision operative, Courses of Action which consists of Strategies, and Tactics, which implement Strategies. Strategies channel efforts towards goals while Tactics channel efforts towards Objectives. The framework also consists of Directives: Business Policies and Business Rules which govern Strategies and Tactics, as well as Influencers, which can influence any concept. Influencers on Ends and Means are classified as internal or external and assessed using the SWOT categorization, classifying each influence as a Strength, Weakness, Opportunity or Threat.

As this framework consists of many concepts which are comparable to those in goal- and agent-oriented frameworks, we look for means of explicit and systematic analysis. Objectives are defined in concrete terms, for example, by January 1, 2005, 95% on-time pizza delivery, and one or more of these objectives are used to measure the satisfaction of goals. Objectives appear to be stated in binary terms, being accomplished or not, but it is not stated whether the combination of multiple objectives to assess the achievement of a goal is performed through an AND or OR combination. Strategies are meant to accomplish goals, and are again described in binary terms, for example, deliver pizzas to the location of the customer's choice. Tactics are specific actions to implement strategies, such as hire drivers with their own vehicles to deliver pizzas. In general, the Framework appears to only consider binary satisfaction: Objectives are satisfied or not, and Strategies and Tactics, if executed, accomplish Objectives and Goals. There is no notion of partial satisfaction or a consideration of non-functional goals, which are difficult to measure in a binary manner. In this way, this approach is similar to a binary AND/OR goal tree, but with additional constructs and considerations. Evaluation for such structures is simple, involving the propagation of binary, yes/no values.

2.6 Alternative Reasoning Approaches

We briefly consider approaches which may be applicable to reasoning over goal models.

2.6.1 Qualitative Reasoning

The AI field of qualitative reasoning has potential application to qualitative goal evaluation. Here, investigations are done to determine what sort of reasoning can be done over continuous variables with little information (Forbus, 1997). Qualitative representations of quantity and state are potentially relevant. Qualitative representations of quantity are typically low resolution (low information) representations over real numbers. In the case of qualitative goal evaluation, we cannot assume that qualitative measures of satisfaction and denial are always abstractions of concrete real numbers. In some cases, satisfaction and denial may be quantifiable, such as for tasks or hard goals, but for softgoals, such as “Job Satisfaction”, quantitative measures may not exist. Even if a quantitative measure for such softgoals could be devised, there is likely no universally agreed upon quantification, only an assortment of measures which act as estimations. State representations partition the behavior of a system into distinct parts. Such concepts do not naturally match with qualitative labels of satisfaction or denial, as the distinction between evaluation labels, especially between full and partial satisfaction (denial), does not necessarily correspond to a distinct behavior of an element. In fact, the notion of an elements “behavior” is vague, and dependent on the contextual nature of the element in question. Generally, although this approach has similarities to the reasoning typically applied to goal models, there are conceptual boundaries which inhibit its applicability.

2.6.2 Multi-Valued Logic

The use of alternative logical representations, such as multi-valued logic, could have potential application to goal model evaluation, especially for the automatic resolution of multiple, incoming softgoal labels (Gottwald, 2000). Such logic could be used to represent the seven possible qualitative evaluation values used in (Horkoff, 2006). However, there may be difficulties in scaling this approach, as softgoals can potentially have many incoming contribution links, and the number of possible combinations of the seven labels would increase exponentially. If automatic resolution of qualitative softgoals is desired, it seems easier to create automatic rules to combine values, as is done in (Amyot et al., 2010).

2.7 General Guidelines for Goal Model Analysis Technique Selection

By examining the capabilities of GORE analysis techniques described in our survey, we produce a list of categories for potential benefits gained through method application, namely: domain understanding, communication, model improvement, scoping, requirement elicitation, requirements improvement, and design. The list of benefits and guiding questions is not meant to be complete, but to act as a useful starting point for understanding the benefits of GORE analysis procedures. Our objective is to provide selection guidance to users of goal models and potential analysis procedures, depending on their objectives and the characteristics of the domain. Our objectives focus on techniques aimed for the analysis of goal models (Section 2.2) as opposed to the construction of goal models (Section 2.4), or analysis of other types of artifacts (Section 2.5).

In order to better motivate the mapping between these benefits and the approaches in our survey, several guiding questions are included with each benefit category, reflecting the capabilities of goal model analysis procedure. Although we provide justification for the mapping, it is often based on our experiences with goal model application, and is meant to provoke useful discussion.

Table 40 lists the categories of GORE analysis benefits, the guideline questions, and recommended procedures depending on the answer to the guideline questions. An interactive version of Table 40, current as per 2010, can be downloaded from:
www.cs.utoronto.ca/~jenhork/GOREAnalysisSelectionTable.zip.

2.7.1 Domain Understanding

All techniques can potentially improve understanding of the domain; however, some procedures have particular qualities which make them especially helpful. Satisfaction analysis techniques can help to explain cause and affect relationships when selecting alternatives. Procedures which explicitly support agent-oriented constructs can help to understand the dynamics of stakeholder relationships at a high or detailed level. Procedures which focus on qualitative evaluation are more appropriate for high-level models, reasoning over non-functional requirements which are difficult to quantify. Such procedures may not provide sufficient granularity at detailed levels. Techniques such as planning, simulation and model checking force the user to add detail to the model which may not be available in early RE; however, adding this detail leads to the discovery of detailed requirements. Using these ideas, we can derive a series of questions concerning high-level or detailed domain understanding which can guide procedure selection.

2.7.2 Communication

Goal models and analysis procedures can be used to communicate domain information, trade-offs, alternative designs, and selection justification. Analysis procedures which provide a justification for their decisions aid in communication. When communicating with stakeholders, the rationale behind results must be easy to understand, especially if stakeholders do not have a technical background. Forward satisfaction techniques help to justify the selection of one alternative over another and can be easy to explain to stakeholders. The results of other techniques may not be as easily explained or justified.

2.7.3 Model Improvement

Although any procedure could be used to improve the quality of the model by prompting users to notice deficiencies in model construction or content, work in this thesis claims that methods which involve human interaction are more likely to cause model changes, as the user is forced to carefully examine propagation in steps through the model. Further work refines this claim, stating that these benefits may be dependent on knowledge of the modeling language or the participation of a modeling facilitator. These theories and results described in more detail in Chapter 12. Automatic evaluation, on the other hand, treats model evaluation as a black box. Model checking procedures explicitly support the ability to check properties over models, potentially improving model quality when desired checks fail. We have classified procedures for model improvement in Table 40, including guideline questions.

2.7.4 Scoping

We hypothesize that agent-oriented procedures are more helpful in supporting analysis in order to determine system and actor boundaries. This is reflected in Table 40.

2.7.5 Requirements Elicitation

The process of finding new high-level requirements is related to improving the accuracy of the model. Interactive procedures force the user to examine the model, finding deficiencies and prompting further elicitation. For the discovery of detailed requirements, procedures which force users to add additional, quantitative, or detailed information to the model can lead to the discovery of new, specific requirements.

Table 40 Mapping of Objectives to GORE Analysis Techniques

Category	Guidelines	Recommended Procedures
Domain Understanding	QU1. Does the domain contain a high degree of social interaction, have many stakeholders with differing goals, or involve many interacting systems?	Yes. Try: Agent Approaches: i*/GRL Satisfaction Analysis (Amyot et al., 2010) (Maiden et al., 2007) i* Metrics (Franch, 2006)(Franch & Maiden, 2003)(Franch et al., 2004) Tropos Metrics, Planning, or Model Checking (Asnar et al., 2007)(Bryl et al., 2006, 2007)(Bryl et al., 2006b): (Fuxman et al., 2003)(Fuxman et al., 2001)(Gans et al., 2002) SNET(Gans et al., 2005)(Gans et al., 2003b)(Gans et al., 2003a)
	QU2. Do you need to understand details of the system at this point? Do you have access to detailed information such as cost, probabilities, and conditions? Can you express necessary or desired domain properties?	Yes. Try: Quantitative or Detailed Information: Tropos Probabilistic Satisfaction Analysis (Asnar & Giorgini, 2006) (Giorgini et al., 2004a) (Giorgini et al., 2004b)(Giorgini et al., 2005) KAOS Satisfaction Analysis (Letier & Lamsweerde, 2004), GRL Quant. Analysis (Amyot et al., 2010), i* Quant. Metrics (Franch, 2006)(Franch & Maiden, 2003)(Franch et al., 2004) Tropos Planning (Asnar et al., 2007)(Bryl et al., 2006, 2007)(Bryl et al., 2006a): Tropos Modeling Checking (Bryl et al., 2006b): (Fuxman et al., 2003)(Fuxman et al., 2001)(Gans et al., 2002) SNET(Gans et al., 2005)(Gans et al., 2003b)(Gans et al., 2003a) (Gans et al., 2003b) i* Simulation(X. Wang & Lespérance, 2001), or Model Checking: Tropos (Bryl et al., 2006b): (Fuxman et al., 2003)(Fuxman et al., 2001)(Gans et al., 2002) SNET(Gans et al., 2005)(Gans et al., 2003a)
Communication	QC1. Do you need to communicate with stakeholders? Validate requirements in the model? Justify recommendations?	Yes. Try: Forward Satisfaction Approaches: NFR(Chung et al., 2000) Tropos (Asnar & Giorgini, 2006) (Giorgini et al., 2004a) (Giorgini et al., 2004b)(Giorgini et al., 2005) KAOS (Letier & Lamsweerde, 2004) i*(Horkoff, 2006)(Maiden et al., 2007) GRL((Amyot et al., 2010)
Model Improvement	QM1. Are you confident in the accuracy, structure, and completeness of domain knowledge and models?	No. Try: Interactive Approaches: NFR (Chung et al., 2000) i* (Horkoff, 2006)(Maiden et al., 2007) Tropos (Asnar et al., 2007)(Bryl et al., 2007) SNET (Gans et al., 2005)(Gans et al., 2003a) i* Metrics (Franch, 2006)
	QM2. Would you like to verify critical properties over the model?	Yes. Try: Model Checking: Tropos (Bryl et al., 2006b) (Fuxman et al., 2003)(Fuxman et al., 2001)(Gans et al., 2002) SNET(Gans et al., 2005)(Gans et al., 2003a)
Scoping	QS1. Do you need to determine system scope?	Yes. Try: Agent Approaches: i*/GRL Satisfaction Analysis (Amyot et al., 2010)(Horkoff, 2006) (Maiden et al., 2007) i* Metrics (Franch, 2006)(Franch & Maiden, 2003)(Franch et al., 2004) Tropos Metrics, Planning, or Model Checking (Asnar et al., 2007)(Bryl et al., 2006, 2007)(Bryl et al., 2006b): (Fuxman et al., 2003)(Fuxman et al., 2001)(Gans et al., 2002) SNET (Gans et al., 2005)(Gans et al., 2003a)
Requirements Elicitation	QE1. Do you need to find more high-level requirements? Are you looking for ways to prompt further elicitation?	Yes. Try: Interactive Approaches: NFR(Chung et al., 2000) i*(Maiden et al., 2007) Tropos(Asnar et al., 2007)(Bryl et al., 2007) SNET(Gans et al., 2005)(Gans et al., 2003a) i* Metrics(Franch, 2006)
	QE2. Do you need to find detailed system requirements?	Yes. Try: Quantitative or Detailed Information: Tropos Probabalistic Satisfaction Analysis (Asnar & Giorgini, 2006) (Giorgini et al., 2004a) (Giorgini et al., 2004b)(Giorgini et al., 2005) KAOS Satisfaction Analysis (Letier & Lamsweerde, 2004) GRL Quant. Analysis (Amyot et al., 2010) i* Quant. Metrics (Franch, 2006)(Franch & Maiden, 2003)(Franch et al., 2004)) Tropos Planning (Asnar et al., 2007)(Bryl et al., 2006, 2007)(Bryl et al., 2006a): Tropos Modeling Checking (Bryl et al., 2006b) (Fuxman et al., 2003)(Fuxman et al., 2001)(Gans et al., 2002) SNET(Gans et al., 2005)(Gans et al., 2003b)(Gans et al., 2003a) (Gans et al., 2003b) i* Simulation (X. Wang & Lespérance, 2001)

	QE3. Do you need to consider non-functional requirements difficult to quantify?	Yes. Try: Approaches supporting softgoals or contributions: NFR(Chung et al., 2000) i* Satisfaction Analysis (Horkoff, 2006)(Maiden et al., 2007) Tropos Satisfaction Analysis (Asnar & Giorgini, 2006) (Giorgini et al., 2004a) (Giorgini et al., 2004b)(Giorgini et al., 2005) Tropos Model Checking(Fuxman et al., 2003)(Fuxman et al., 2001) GRL(Amyot et al., 2010) i* Metrics(Franch, 2006)(Franch & Maiden, 2003)(Franch et al., 2004) SNET(Gans et al., 2005)(Gans et al., 2003b)(Gans et al., 2003a)
	QE4. Do you need to capture domain assumptions?	Yes. Try: Approaches using Satisfaction Arguments: i* Satisfaction Arguments (Maiden et al., 2007)
Requirements Improvement	QR1. Are you working with a system where safety/security/ privacy/risks or other specific properties are critical considerations?	Yes. Try: Analysis over Specific Constructs or Metric Approaches: KAOS(Letier & Lamsweerde, 2004) i* Metrics(Franch, 2006)(Franch & Maiden, 2003)(Franch et al., 2004) AGORA(Kaiya et al., 2002) Tropos Risk, Trust, and Security(Asnar & Giorgini, 2006)(Asnar et al., 2007) (Bryl et al., 2006b): (Gans et al., 2002) SNET Trust(Gans et al., 2003b)
	QR2. Do you need to find errors and inconsistencies in requirements?	Yes. Try: Model Checking: Tropos(Bryl et al., 2006b): (Fuxman et al., 2003)(Fuxman et al., 2001)(Gans et al., 2002) SNET(Gans et al., 2005)(Gans et al., 2003a)
Design	QD1. Are you aware of a sufficient number of high-level design alternatives?	No. Try: Agent, Planning, Forward and Backward Satisfaction Approaches: NFR(Chung et al., 2000) i* Satisfaction Analysis (Horkoff, 2006)(Maiden et al., 2007) Tropos Planning(Asnar et al., 2007)(Bryl et al., 2006, 2007)(Bryl et al., 2006a): KAOS(Letier & Lamsweerde, 2004) GRL Forward Satisfaction Analysis(Amyot et al., 2010) SNET Planning(Gans et al., 2005)(Gans et al., 2003a)
	QD2. Are you aware of a sufficient number of detailed design alternatives?	No. Try: Quantitative Planning, Forward and Backward Satisfaction Approaches: KAOS Satisfaction Analysis (Letier & Lamsweerde, 2004) GRL Forward Satisfaction Analysis (Amyot et al., 2010) Tropos Planning(Bryl et al., 2006, 2007) SNET Planning(Gans et al., 2005)(Gans et al., 2003a)
	QD3. Do you need to evaluate and choose between high-level design alternatives?	Yes. Try: Satisfaction Analysis, Metrics and Agent Approaches: KAOS Satisfaction Analysis(Letier & Lamsweerde, 2004) i* Forward Satisfaction(Horkoff, 2006)(Maiden et al., 2007) GRL Satisfaction Analysis(Amyot et al., 2010) i* Metrics(Franch, 2006)(Franch & Maiden, 2003)(Franch et al., 2004)) Tropos Risk(Asnar et al., 2007)
	QD4. Do you need to evaluate and choose between detailed design alternatives?	Yes. Try: Quantitative or Detailed Information: Tropos Probabalistic Satisfaction Analysis (Asnar & Giorgini, 2006) (Giorgini et al., 2004a) (Giorgini et al., 2004b)(Giorgini et al., 2005) KAOS Satisfaction Analysis (Letier & Lamsweerde, 2004) GRL Quant. Analysis (Amyot et al., 2010) i* Quant. Metrics (Franch, 2006)(Franch & Maiden, 2003)(Franch et al., 2004) Tropos Planning (Asnar et al., 2007)(Bryl et al., 2006a, 2007)(Bryl et al., 2006b): Tropos Modeling Checking (Bryl et al., 2006b) (Fuxman et al., 2003)(Fuxman et al., 2001)(Gans et al., 2002) SNET(Gans et al., 2005)(Gans et al., 2003b)(Gans et al., 2003a) i* Simulation(X. Wang & Lespérance, 2001)
	QD5. Do you need to find acceptable processes?	Yes. Try: Planning Approaches: Tropos Planning(Asnar et al., 2007)(Bryl et al., 2006a, 2007)(Bryl et al., 2006b): SNET Planning(Gans et al., 2005)(Gans et al., 2003a)
	QD6. Do you need to test run-time operation before implementation?	Yes. Try: Simulation Approaches: SNET(Gans et al., 2005)(Gans et al., 2003b)(Gans et al., 2003a) i* Simulation(X. Wang & Lespérance, 2001)

When considering non-functional requirements that are difficult to quantify, such as privacy or customer satisfaction, support for softgoal or contribution notations are critical. The procedure by (Maiden et al., 2007) explicitly asks users to capture domain assumptions associated with system requirements in textual arguments associated with model evaluation.

2.7.6 Requirements Improvement

After an initial set of requirements has been captured, the requirements can be improved via checks for consistencies or errors or consideration of critical properties. Procedures which support checks over specific properties like safety and security are particularly applicable. Model checking approaches are specifically targeted to finding errors and inconsistencies in requirements captured in goal models.

2.7.7 Design

Once a set of requirements has been captured in the model, the models can be used to find and evaluate high-level or detailed alternative design solutions. Planning procedures find acceptable plans (design alternatives). Backward analysis procedures find a set of acceptable options, given desired goal satisfaction levels. These procedures can only find alternatives already in the model, while approaches for forward satisfaction explicitly encourage users to brainstorm for new alternatives when goals are not sufficiently satisfied.

Forward satisfaction analysis procedures are specifically aimed to evaluate design alternatives by marking selected alternatives as satisfied in the model. Similarly, simulation procedures simulate specific scenarios or alternatives. To a certain degree, metric and model checking procedures can also be used to evaluate alternatives, by creating and evaluating alternative models. The distinction between high-level and detailed design alternatives is similar to the distinction between high-level and detailed domain understanding; with agent-oriented procedures more helpful for high-level understanding and quantitative or detailed information procedures more helpful for detailed design.

2.8 Guideline Usage Examples

We apply our guidelines to two of the case studies, the Wireless Service described in (Amyot et al., 2010) and the Counseling Service described in Section 1.1.

2.8.1 Wireless Service

In this example, a new wireless service must be added to an existing network, and the analysts must decide where the service and its data are to be located. Options include the data in service control point, data in new service node, service in central switch or service in service control point. These alternatives produce various effects on the goals of the service provider, and produce different requirements for service vendors.

This particular domain contains a few interacting systems (service provider, vendor, and the wireless system provider) (QU1). The analysts/modelers do not yet understand the details and do not have access to specific information to formulate and check specific desired properties (QU2). There is no mention of a need to communicate with stakeholders (QC1). The domain is relatively well understood, the scope is clear, knowledge and models seem sufficiently complete (QE1, QS1, QM1). Several non-functional requirements such as low cost and high performance must be considered (QE3). There is no mention of the need to capture domain assumptions (QE4). In considering important properties, data privacy is an important consideration in wireless networks (QR1). The example does not yet have enough information to run formal checks for consistency over the model (QR2). The analyst is aware of the high-level alternatives, but need to discover which high-level alternative works the best (QD1, QD3). Finally, the example description does not express a need to get into detailed design alternatives, find processes, or simulate operation (QD2, QD4, QD5, QD6).

Recommendations. Our guidelines suggest the use of agent-oriented approaches supporting softgoals to consider the social nature of the problem, along with satisfaction analysis or metrics to select a high-level alternative, i* Satisfaction Forward Analysis (i* evaluation, Horkoff, 2006), GRL Satisfaction Analysis (Amyot et al., 2010), Tropos Risk Analysis (Asnar & Giorgini, 2006), and/or i* Metrics (Franch, 2006; Franch et al., 2004; Franch & Maiden, 2003). The satisfaction analysis and metric techniques could be repeated or adjusted to specifically support privacy analysis.

2.8.2 Online Counseling

An organization providing free counseling services for kids and youth would like to provide services online. However, they must continue to satisfy their key requirements of privacy and confidentiality, while maintaining a high quality of counseling, sufficient funding, and happy counselors.

In this example there is a high degree of social interaction; we need to consider the organization, counselors, youth, the general public, etc (QU1). The analyst/modeler in the example does not yet understand the details and the stakeholders are not aware of such specific information (QU2). Communication with stakeholders is important, we need to explain our criteria and justify our design selections (QC1). Because of the unfamiliarity of the domain, analysts are not confident in the accuracy or completeness of our models (QM1). The scope is difficult to determine, it is hard to know what to include in the models (QS1). In this case, many non-functional requirements such as helping youth and counselor job satisfaction must be considered, and it would be helpful to capture assumptions about the domain (QE3, QE4). In this example, privacy and anonymity of youth information is critically important (QR1). The example describes an interest in finding a variety of high-level counseling alternatives (chat room, bulletin board, wiki, etc), and evaluating their effectiveness in the model (QD1, QD3). It may be useful to find the most successful process for counseling online and it would be nice to explore the throughput of the system in terms of responses to kids and counselor backlog (QD4, QD5).

Recommendations. Our guidelines suggest use of interactive, agent-oriented techniques for forward satisfaction analysis supporting softgoals in order to learn about the domain, find high-level design alternatives, and communicate with stakeholders, i* Satisfaction Analysis (i* evaluation, Horkoff, 2006). In further steps, models could be analyzed for anonymity or privacy with the same techniques or with GRL Satisfaction Analysis (Amyot et al., 2010), and/or i* Metrics (Franch, 2006; Franch et al., 2004; Franch & Maiden, 2003). If the required detailed information is available, Tropos planning techniques could be used to find plans (Asnar et al., 2007; Bryl et al. 2006a; 2009a), while other approaches could be used to simulate a process, SNET (Gans et al., 2003a; Gans et al., 2002; Gans et al., 2005) or i* Simulation (X. Wang & Lespérance, 2001).

2.9 Conclusions

In this chapter, we have reviewed and assessed existing goal-oriented modeling and analysis techniques, including techniques which propagate satisfaction values, calculate metrics, find acceptable models using planning algorithms, simulate model behavior, and check formal properties. We have also given a brief overview of related approaches in the fields of Requirements Engineering and Business, and have reviewed alternative means of supporting and reasoning over design decisions.

The diversity of goal model analysis techniques creates a barrier for adoption of such techniques in practice. This chapter has enumerated potential benefits of goal model analysis and provided initial guidelines for choosing techniques to meet these objectives. The guidelines were illustrated with several examples.

We continue our analysis of related work in the next two chapters. Chapter 3 enumerates requirements for early RE agent-goal model analysis and then considers the appropriateness of existing goal model analysis procedures for early RE, while Chapter 4 provides a detailed comparison of forward analysis procedures in order to evaluate their analysis power.

Chapter 3 Requirements for Early RE Agent-Goal Model Analysis

In this chapter, the challenges of early requirements analysis are used to motivate the use of agent-goal models as a basis for the early RE Analysis Framework introduced in this work. Challenges specific to agent-goal model analysis for early RE introduced in Chapter 1 and gathered from our Chapter 2 review are reconsidered in order to produce a list of requirements for early RE agent-goal model analysis. These requirements are used to evaluate the suitability of procedures surveyed in Chapter 2 for analysis in early RE. This evaluation is used to select procedures to be included in and expanded upon in the framework produced in this thesis.

3.1 Use of Agent-Goal Models for Early RE Analysis

Chapter 1 outlines several challenges in the elicitation, capture and analysis of system requirements in the early or initial stages of RE. We return to these challenges in an effort to identify suitable techniques or methods for application in early RE.

Early stages of system analysis involve recognizing and understanding many complex aspects, including stakeholders, *stakeholder needs*, existing systems, interactions, and solution alternatives. This complexity calls for a means to *abstract* away less important detail, or to create views focusing on central concepts and relationships. Such abstractions or views can help analysts, stakeholders and other parties to *communicate* about the system domain, sharing their perspectives and areas of focus. Communication amongst key parties leads to a *convergent understanding* of system entities and goals. The *involvement of key stakeholders* in the process of elicitation, communication and understanding is critical to obtain a sufficient understanding of the requirements.

The process of eliciting, abstracting, and building a consensus on stakeholder needs is complicated by *incomplete domain information* in the early stages of analysis. Although elicitation aims to create a complete understanding of the domain, the inherit complexities of a socio-technical system mean that not all entities, goals, or relationships may be known, especially given time pressures on the analysis process. An implicit tradeoff exists between the completeness and complexity of analysis, with abstraction aiming to deal effectively with incompleteness by focusing on key system entities, relationships and needs.

In addition, even if key needs are known, it may be difficult in the early stages to explicitly define important objectives. Stakeholders may be able to identify key needs such as profit, customer satisfaction, increased market share, or system security, but it may be *difficult to come up with formal or quantitative measures* for such objectives in early analysis. Despite the difficulties to completely or concretely understand a complex domain, *key decisions* concerning project scope, focus and functional alternatives are made during early requirements analysis. We summarize the challenges of early RE analysis in the following list:

- *Capturing stakeholder needs*
- *Abstracting complex domains*
- *Communicating understanding*
- *Building convergent understanding*
- *Involving key stakeholders*
- *Incomplete domain information*
- *Difficult to come up with formal or quantitative measures*
- *Making key decisions with incomplete or imprecise information*

Several techniques or methods could be applicable to meet the early RE challenges of *abstraction*, *communication*, and *convergent understanding*. The most common methods for capturing and analyzing information elicited in early requirements involve some form of free or structured text or tables. For example, a traditional requirements specification, or a method organizing requirements into structured text templates, such as the Volere Specification Method (S. Robertson & J. Robertson, 2006). Although such approaches allow for the flexibility of natural language, facilitating stakeholder participation, they lack the ability to facilitate visual abstraction and contain the ambiguities of natural language which may impede convergent understanding.

Using models to capture and explain results of early elicitation can better facilitate abstraction, can help to easily communicate a particular point of view, and can achieve a convergent mental picture of the entities, relationships and issues in a system domain. Although we focus on the use of models and their ability to facilitate analysis in this work, an effective requirements engineering process would use a combination of artifacts and approaches, (e.g., models, text, and tables), and would use a variety of interaction techniques to encourage adequate understanding and stakeholder participation, (e.g., meetings, focus groups, and surveys).

In addition to aiding abstraction and convergent communication, models can make it easier to get *key stakeholders involved* in the requirements process. Working over a concrete artifact, like a model, can help to focus elicitation and analysis, and can help to show progress and create a sense of accomplishment. Although similar benefits may be achieved via textual lists, models can provide more effective visual aids for relating concepts together, or for abstracting away detail.

Although the use of models helps to address some of the challenges of early RE (abstraction, communication, convergent understanding, and the involvement of stakeholders), the choice of what type of models to use makes a significant difference in addressing remaining challenges. Models typically used in requirements and analysis and software design, such as various UML models, ERDs, DFDs, or SADTs are able to effectively represent the domain entities, relationships, and behaviors. They help to answer questions such as "what?", "how?", and "when?" Although these models can be useful in early RE, they fail to capture and help modelers understand *stakeholder needs*. These models are also not specifically intended to help with early *decision making*, allowing for "what if?" questions. Static and dynamic models can help show the differences between system alternatives to a certain extent, by showing different models for different alternatives, or by showing different functional paths in a scenario, but they lack the ability to show the impact of alternatives on system objectives.

Agent-goal models are able to capture stakeholder and system objectives effectively (van Lamsweerde, 2001), mapping them to "agents", particular people, roles, or systems. Agent-goal models allow for *incompleteness*, modeled concepts can make sense even if they domain is not completely represented. The presence of softgoals allow for the explicit consideration of important objectives which are *difficult to formalize or quantify* in early analysis. The structure of goal models allows for the representation of alternatives linked to goals, facilitating "what if?" analysis from the viewpoint of domain objectives.

Other modeling approaches could also be used to capture system objectives. For example, the Soft System Methodology approach uses rich pictures, drawings and sketches to capture the domain. This approach is aimed at dealing with systems where objectives are difficult to clearly define and are often conflicting (Checkland, 2000). Such models may or may not include user goals, depending on the nature of the resulting sketches. Although the lack of defined syntax for such models allows for flexibility, it does not enable "what if?" analysis using model structure. Other types of models, such as argument maps, could be used to aid early decision making (Gelder, 2009). However, such models typically do not contain goal-oriented concepts and are more suited to describing an argument and capturing rationale than for asking "what if?" Goal models which do not support agent concepts, such as models in the NFR Framework (Chung et al., 2000), could be applied to address early RE challenges. However, it is useful not only to elicit domain objectives, but to capture the source of objectives, facilitating traceability and analysis from different perspectives.

As a result of these considerations, we focus on the use of agent-goal models for early RE analysis. Such models allow users to create abstractions of stakeholders and their needs. They are able to provide an effective tradeoff between the expressiveness required for systematic analysis facilitating early decision making and the flexibility required for expressing high-level concepts in early RE.

A successful requirements analysis process may make use of several types of models, capturing many views of the system. For example, in the RESCUE method, i* (agent-goal) models are checked against Use Cases and Human Activity models for completeness (Maiden, Jones, Manning, Greenwood, & Renou, 2004) (see Section 2.3.6 for further examples). In this work, we focus on the iterative analysis capabilities of agent-goal models. Further work could link the agent-goal models and processes used in this framework to additional, useful modeling or textual artifacts.

3.2 Requirements for Early RE Agent-Goal Model Analysis

Chapter 1 has described challenges in analyzing agent-goal models in early RE. We have surveyed existing approaches to goal model analysis in Chapter 2. We decompose the identified challenges in early RE, using requirements identified in existing in exiting work, and our own experience using agent-goal model analysis in practice, to derive requirements for a framework aimed for early RE agent-goal model analysis. Early RE agent-goal model challenges are summarized in the following list:

- **Model complexity**
- **Model completeness**
- **Model accuracy**
- **Domain Knowledge**
- **Model interpretation**
- **Model Flexibility**
- **Decision rationale**
- **Stakeholder Involvement**
- **Analysis power**
- **Procedure usability**
- **Procedure Selection**

Model Complexity. Early RE agent-goal models cover complex social situations, and can often become large and complicated (see Figure 1 for an example large model). From this challenge, we can derive requirements for scalability, comprehension, and tool support, as follows:

R1 Scalability: The analysis framework must contain techniques which are applicable over large models.

R2 Analysis Comprehension: The analysis framework should contain methods to support comprehension of analysis results over complex models.

R3 Partial Automation: Procedures in the analysis framework should be supported by tools which provide some automation for analysis over large models.

The introduction of R1 and R3 requires a discussion of what is meant by "large" for early RE models. Generally, the size of the model is constrained by the cognitive ability of modelers, as these models are always created manually as part of an elicitation process. We further explore the concept of "large" early RE models and other scalability issues in Section 11.4.

Model Completeness & Accuracy. It is difficult to create early RE models which are complete, instead the aim is for relative completeness, or complete enough to facilitate useful analysis. However, it is difficult to know when a model is sufficiently complete or accurate. Frameworks which support agent-goal early RE analysis should contain methods which aim to increase the completeness and accuracy of models. We argue that analysis procedures which prompt iteration over models helps to increase the completeness and accuracy of models, provoking model changes until the model reaches a relatively stable state. We also claim that the act of noticing model incompleteness or inaccuracies is greatly enhanced when analysis is interactive, prompting modelers to examine contentious areas of the model. The validity of these claims is examined further when describing framework case studies in Chapter 12. From these claims, we derive the following requirements.

R4 Model Iteration: The framework should contain analysis methods which encourage model iteration.

R5 Interactive Procedures: The framework should contain analysis methods which encourage interactive analysis.

Domain Knowledge: The analysis procedures in the framework prompt an increase in domain knowledge.

R6 Prompt Further Elicitation: The framework should contain analysis methods which reveals gaps in domain knowledge and prompts users to fill those gaps through further elicitation.

Model Interpretation: The flexibility and inexpressiveness of agent-goal models can lead to divergent interpretations of their syntax. Ideally, frameworks for agent-goal model analysis should address this challenge:

R7 Definition: The framework should contain a more formal or precise definition of the underlying agent-goal model framework. The analysis procedures should be formally defined in an effort to avoid divergent application or interpretation.

Model Flexibility: The framework should allow analysis over high-level, potentially ambiguous concepts which are not represented formally or quantitatively.

R8 Accommodate Inexpressiveness: The framework should contain analysis procedures which do not require formal or quantitative definitions of model contents.

R9 Accommodate High-Level Domain Information: The framework should contain analysis procedures which do not require detailed domain information, difficult to acquire in early RE stages.

Decision Rationale. As early analysis often involves key decisions made over incomplete and imprecise information, it is important to capture the rationale for decisions made, specifically linking this rationale to domain goals. In agent-goal models, decisions can be made at the model level, making human judgments over contentious areas of the model, or can be made between high-level domain alternatives using analysis results over the model. From these challenges, we derive the following requirements:

R10 Human Judgments: The framework should support ways to capture, store and analyze the analysis decisions made over contentious areas in the model.

R11 Decision Rationale: The framework should support ways to capture the rationale for decisions amongst alternatives, including varying analysis results which lead to these decisions.

Stakeholder Involvement. Early RE analysis should encourage the involvement of key stakeholders for elicitation and validation of early requirements. The framework requirements for model iteration (R4) and an interactive procedure (R5) already help to address these challenges by encouraging stakeholders to interact with the analysis process, iterating over the model. We further address this challenge by adding a requirement concerning a framework methodology:

R12 Iterative Methodology: The framework should contain clear methodologies to guide the process of interactive analysis including iteration over model content.

We distinguish between R4 Model Iteration and R12 Iterative Methodology by calling for analysis algorithms which explicitly encourage model changes and then explaining how these procedures could be used as part of a modeling and analysis method, supporting iteration between modeling and analysis or different types of analysis.

Analysis Power. Analysis procedures for early RE analysis should support a variety of types of analysis, allowing user to ask several types of questions over the model. Although there is tradeoff between analysis capabilities and the need for more detailed model information, early analysis should at least facilitate "What if?"-type questions.

Analysis procedures should produce accurate, sensible and reliable results. Determining the accuracy of results which analyze the effectiveness of alternatives in the "to-be" domain is difficult. We can consider "sensible" results to be results that are generally agreed upon by stakeholders, which seem sensible given domain knowledge. In the absence of reliable methods to test the accuracy of analysis results, we can test that they produce reliable results by making use of the presence of multiple, similar analysis approaches in order to compare results across procedures.

R13 Analysis Questions: The framework should support a variety of analysis questions over agent-goal models, including "What if?" analysis.

R14 Reliable Analysis: The framework should produce results which are accurate, sensible and reliable.

Procedure Usability. Although several existing agent-goal model analysis procedures may be applied in an early RE context, it is not clear if these procedures are practically usable. To ensure usability, analysis procedures should be as simple as possible to apply, with as much complexity as possible hidden by tool support, and should be guided by clear methodologies (R11).

R15 Simple Analysis Procedures: The framework should contain analysis procedures which are simple enough to be applied with minimal training.

R16 Tool Support Hides Complexity: The implementation of analysis procedures in the framework should encode and hide as much complexity as possible from the user

R15 raises questions concerning “simple enough” and “minimal training”. These questions will be addressed as part of the scalability tests described in Section 11.4.

Procedure Selection. The presence of many analysis procedures for agent-goal Models makes it difficult for potential users to select an existing analysis procedure. Depending on domain factors such as available information, stakeholder time, and level of safety criticality, different procedures may be more or less appropriate. Although this challenge does not lead to requirements specifically for the framework in development, it does call for a review and assessment of existing agent-goal analysis procedures. Such a review should include guidelines for application based on domain-dependent factors. We address this challenge via surveys and assessments in Chapter 2 and Chapter 4.

The requirements for our early RE agent-goal model analysis framework are summarized in the following list:

- **Model complexity**
    - **R1 Scalability**
    - **R2 Analysis Comprehension**
    - **R3 Partial Automation**
- **Model completeness & Accuracy**

  - **R4 Model Iteration**
  - **R5 Interactive Procedure**
- **Domain Knowledge**
  - **R6 Prompt Further Elicitation**
- **Model interpretation**
  - **R7 Definition**
- **Model Flexibility**
  - **R8 Accommodate Inexpressiveness**
  - **R9 Accommodate High-Level Domain Information**
- **Decision rationale**
  - **R10 Human Judgments**
  - **R11 Decision Rationale**
- **Stakeholder Involvement**
  - **R12 Iterative Methodology**
- **Analysis power**
  - **R13 Analysis Questions**
  - **R14 Reliable Analysis**
- **Procedure usability**
  - **R15 Simple Analysis Procedures**
  - **R16 Tool Support Hides Complexity**
- **Procedure Selection**

We summarize the requirements for analysis of agent-goal models in early RE by adding to our goal model from Figure 3. Figure 4 shows the decomposition of our challenge goals into the corresponding requirements. Figure 5 shows some of the perceived conflicts and synergies amongst these goals.

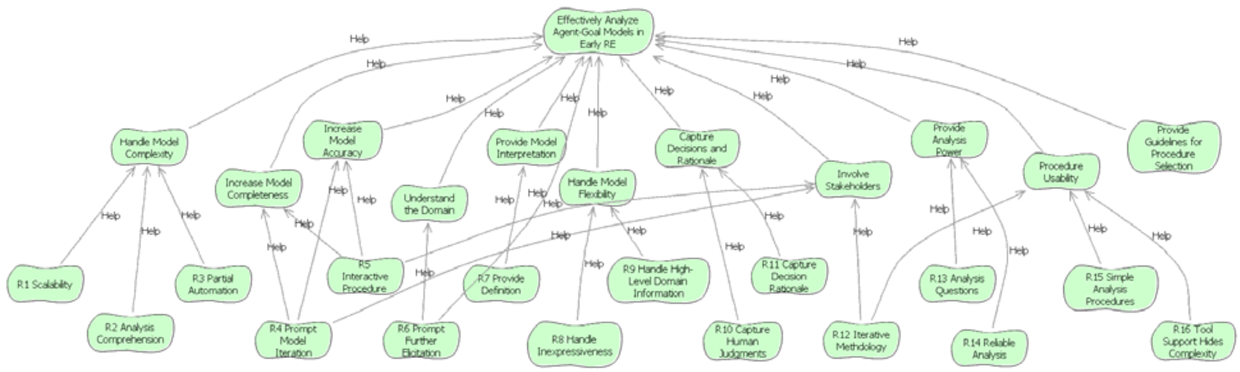


Figure 4: Summary of the Requirements for analysis of Agent-Goal Models in Early RE

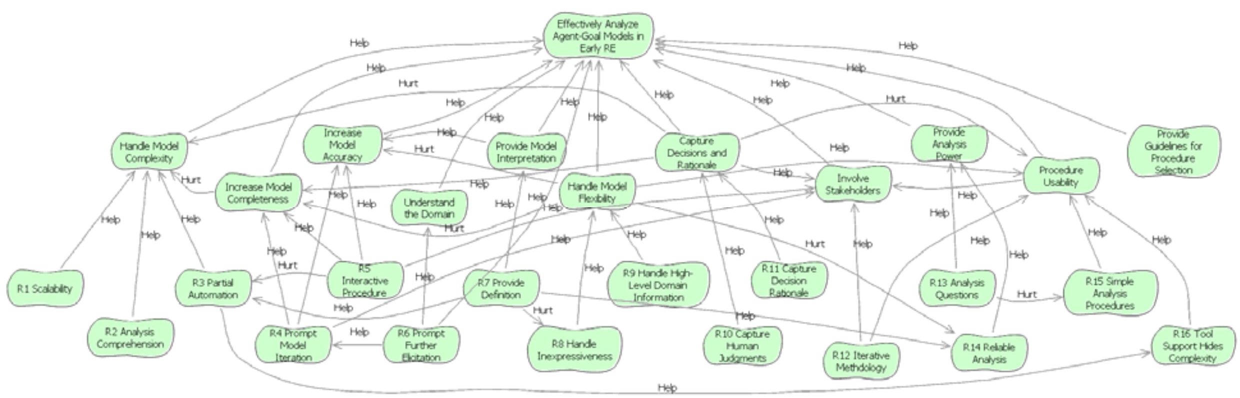


Figure 5: Summary of the Requirements for analysis of Agent-Goal Models in Early RE including Conflicts and Synergies

The requirements derived in this chapter will be used to assess and summarize the contributions of each of the remaining chapters of the thesis. Chapter 13 will assess the overall contributions of the framework in light of these requirements.

3.2.1 Model vs. Method vs. Algorithm Iteration

As several of the challenges and derived requirements are iterative in nature, we include a short description of different types of iteration referred to in this work. The emphasis of requirement R4 is on model iteration, prompting changes to the model which improve accuracy and completeness. Because we aim to prompt this type of iteration with the analysis framework introduced in this work, we require a methodology that takes this iteration into account, guiding users on how to be open to potential changes in the model, and how to iterate over method steps in order to make these changes as part of analysis process. This requirement is captured in R12 Iterative methodology.

There is a third type of iteration which we do not include explicitly in our requirements: algorithmic iteration. An iterative algorithm could be considered any algorithm which has in iterative loop in it (like a for or while loop), any algorithm which repeats, or, more specifically, any algorithm that repeats or iterates, using results from the previous iteration in order to find an increasingly better solution to a problem. An example of the latter would be calculating the root of an equation (x, such that f(x) = 0), which can be done using an iterative algorithm which converges towards the root. In our case, such algorithms would analyze models in an iterative process, reusing previous results in some way, in order to converge on model results. Although iterative algorithms could contribute toward requirements R3, R12 and R13 concerning tool support, they involve choices in implementation, and are not made direct requirements.

The algorithms introduced in this work are iterative in that they repeat, but not purely iterative in that they do not yet completely reuse the results from the previous iteration. More details are provided in Chapter 11.

3.3 Suitability of Existing Goal Model Techniques for Early RE Analysis

Chapter 2 has provided a survey of existing work in goal model analysis. In this section, we evaluate this work in light of our requirements for early RE agent-goal model analysis. This evaluation is used to select procedures to be included in and expanded upon in the framework produced in this thesis. Specifically, we aim to answer the following question:

- Early Analysis: Which techniques may be appropriate for early RE analysis?

In Section 2.7, we have created an initial list of objectives which may be met by goal model analysis techniques throughout the software lifecycle. We now consider techniques for goal model analysis, construction, and consistency in light of the requirements for our early RE agent-goal model analysis framework, as described in this chapter. In other words, we broaden our consideration of goal model techniques to include techniques for goal model construction and inter-model consistency checking, while narrowing our consideration of purpose to early RE analysis. This subsection aims to use our list of early RE agent-goal model analysis requirements to point out gaps in existing goal modeling work. The remainder of the thesis focuses on how our early RE agent-goal model analysis framework will fill these gaps, producing a framework more suitable for early RE analysis.

In Section 3.1, we have justified the use of models, and specifically of agent-goal models in our Framework. Thus, when analyzing the suitability of existing approaches for early RE analysis, we focus only on those techniques using goal models (Section 2.3).

3.3.1 Model Complexity

R1 Scalability: *The analysis framework must contain techniques which are applicable over large models.*

Several approaches for goal model analysis address scalability. For example, approaches for automatic forward and backward analysis associated with the Tropos methodology (Giorgini et al., 2002; 2004; Giorgini, Mylopoulos, & Sebastiani, 2004) address scalability by testing the implementation of their analysis methods on large (thousands of goals), randomly created goal graphs, demonstrating a reasonable (less than five seconds) running time. Although tests for computational run time are useful, this body of work does not address the scalability of manually inputting initial values or targets into large models or into scalability of analysis results comprehension. Although the procedures may run relatively quickly over models with thousands of goals, it is likely that users may have difficulty creating, posing analysis questions, or understanding analysis results over models of this size.

Similarly, work in (Fuxman et al., 2003) reports running time for property and consistency checks over i* models which convert into a large state space. This approach attempts to further address scalability by providing partial automatic translation from i* to Formal Tropos, as the authors recognize that this is a labor intensive process. Although this is a helpful contribution, part of the conversion must still be done manually, which makes this approach difficult to scale.

In addition, the running time for their example models can be as high as 2837 (~47 minutes), with some example checks not returning results, as they exceed the bounds for the model checker.

Approaches introducing qualitative evaluation for i* models (e.g., Liu et al., 2003) argue for scalability via their use of the Telos Framework (Mylopoulos, Borgida, Jarke, & Koubarakis, 1990) to represent the semantics of i*. In other words, instead of using complex graphical representations, reasoning can occur over textual representations of the model. These approaches do not consider the computational complexity of their algorithms. Although the use of an underlying textual representation helps to avoid graphical complexity, the underlying complexity of the model domain is not reduced. Therefore, this approach suffers from similar issues in forming and understanding analysis questions over complex models.

Many goal model techniques focus on application to realistic case studies as a means of testing scalability. For example, work by Maiden et al. (2004) describes a cross-model synchronization process using large models from an industrial case study. Work in (Horkoff, 2006) pushes the boundaries of manually constructed and analyzed models by providing examples of manual analysis over i* models with hundreds of intentions. Models in this work came from a requirements analysis of a social service organization (Easterbrook et al., 2005). Although modeling and analysis were successfully executed in both cases, the processes were cognitively challenging, and required modelers with much expertise in the modeling syntax and domain. Further work should aim to make analysis over larger models more accessible, in part by making complex analysis results more comprehensible, as explored in the following requirement.

R2 Analysis Comprehension: *The analysis framework should contain methods to support comprehension of analysis results over complex models.*

Coverage of this requirement has been addressed in the previous section. Although some existing methods account for computational scalability, agent-goal model analysis approaches generally do not consider the ability of stakeholders to comprehend analysis results – over either simple or complex models. While several methods make use of industrial case studies to test the utility of their approaches, the reaction of stakeholders to realistic model analysis results are typically not described, with model analysis performed by researchers.

The implementation of GRL analysis in jUCMNav, (JUCMNav, 2011), used by (Amyot et al., 2010), (Pourshahid et al., 2008), and, (Pourshahid et al., 2011) uses colors to express the satisfaction values of intentions. Specifically, fully denied is bright red, fully satisfied is bright green, and other values are shades ranging from green to orange to red. Although this implementation is promising, explicit tests of the comprehension gained by this visualization have not been performed, although the tool has been used in several industrial case studies.

In (Horkoff, 2006), evaluation results for optional features derived from a complex case study model were converted to an ordinal scale before being presented to stakeholders for validation. The model itself was deemed too complex to show to stakeholders, and therefore it was assumed analysis results over the model were not comprehensible by non-modelers. Work in this thesis tests these assumptions, applying visualization techniques to aid analysis comprehension (Section 8.2).

R3 Partial Automation: *Procedures in the analysis framework should be supported by tools which provide some automation for analysis over large models.*

We have identified the need for at least partial automation in order to deal with complex models. All approaches for analysis via satisfaction propagation can be at least partially automated, although some of them have been introduced without explicit mention of tool support (e.g., Chung et al., 2000). In fact, several analysis approaches are fully automated, given the model and initial analysis values, (e.g., Giorgini, Mylopoulos, & Sebastiani, 2004; Letier & Lamsweerde, 2004). These procedures provide full automation at the expense of having interactive procedures (R5) and possibly model iteration (R4).

3.3.2 Model Completeness and Accuracy

R4 Model Iteration: *The framework should contain analysis methods which encourage model iteration.*

While several analysis techniques incorporate iteration, the focus of the iteration is not on iteration over the contents of the model based on analysis results, aiming for improvements in model completeness and accuracy. Most iteration occurs via either iterative methods or iterative algorithms. For example, approaches for Tropos evaluation include iteration over target analysis values. If the procedure is unable to produce a solution given the initial target values in backward (top-down) analysis, the authors recommend relaxing target values in an iterative process (Giorgini et al., 2004b). Procedures for i* analysis recommend iteration as part of model construction (Liu, Yu, & Mylopoulos, 2003) or iteration over analysis results in order to find the most satisfactory solution (Liu & Yu, 2004). These procedures do not consider iteration over the model itself as a consequence of analysis. Other methods aimed for model construction include iteration over the model during construction, but do not explicitly recommend iteration over the model as a result of analysis (e.g., Grau & Franch, 2007; Grau et al., 2005a; Grau et al., 2008).

There are several methods which mention model iteration as part of their methodology (e.g., (Bryl et al., 2006a; Fuxman, Liu, Pistore, Roveri, & Mylopoulos, 2003; Wang & Lespérance, 2001). These methods recommend model iteration when no satisfactory analysis results can be found, or when analysis reveals an error. However, these procedures do not go into model iteration as a result of analysis in any depth. They do not offer guidance on what sorts of errors to look for, the type of changes to be made, or in general, how to use analysis results to make beneficial changes to the model. The approaches do not consider how often or with what likelihood model changes will occur, whether the changes are beneficial, or any other aspects which may affect model iteration (e.g., modeler expertise). In these procedures, model iteration is treated as a side effect of errors or inadequacies and not as a desired outcome of the analysis process in order to improve model quality in early RE. Work in (Liaskos et al., 2011) specifically addresses model iteration as a positive benefit of iteratively applying planning and analysis over prioritized preferences, providing an example of how preferences can be gathered iteratively. However, this work does not investigate the conditions (modeler experience, buy-in) needed to provoke model iteration. We include more details about how iteration fits into these procedures in our consideration of R11: Iterative Methodology.

Our previous work in (Horkoff, 2006) examines the ability of the i*, qualitative forward evaluation procedure to prompt improvements in the model. This procedure is expanded and included in Chapter 5 of this work. Horkoff (2006) claims that evaluation improves model quality by improving the accuracy and comprehensibility of models. Model evaluation is likely to provoke changes in the structure of models which improve comprehensibility, specifically a rearrangement of certain syntax constructs to remove ambiguity revealed through analysis. For example, the removal of cycles, splitting of joined links, etc. Furthermore, evaluation may provoke model changes which improve accuracy. Horkoff provides a list of semantic issues which may become apparent through evaluation, such as missing links, redundant goals, and model flow (inappropriate roots and leaves). The work illustrates each type of syntax and semantic change using examples from case studies (these case studies will be summarized briefly in Chapter 12).

Although this work begins to explore the relationships between analysis and iteration, it is limited in that model iteration in case studies is tested only using one expert modeler (the author). The work only tests the iterative power of forward and not backward analysis. The framework developed in this thesis further articulates and tests hypothesis concerning model iteration and improvement, specifically developing and executing several user studies, described in Chapter 12.

R5 Interactive Procedure: *The framework should contain analysis methods which encourage interactive analysis.*

All procedures are at least somewhat interactive in that the user must interact in order to encode an analysis question or scenario in some form, and then interact with the output in order to map the meaning of the results back to the domain.

Several of the surveyed techniques involve interaction of some type, while others are fully automated. See the 4th column of Table 38 for a summary of which techniques involve some sort of stakeholder interaction. Some of the reviewed satisfaction analysis techniques encourage interaction during the analysis process by asking users to make judgments over conflicting or incomplete analysis results (e.g., Chung et al., 2000) or to rewrite satisfaction arguments (Maiden et al., 2007). Others involve users in the procedure input or output by asking them to make judgments over metrics (Franch, 2006), to evaluate candidate plans (Bryl et al., 2009c), to check actor delegations (Asnar et al., 2007), or to invoke actions (Gans et al., 2002). Approaches such as (Asnar et al., 2007; Bryl et al., 2009c; Franch, 2006) aim less at encouraging interaction and more on using stakeholder expertise to initiate analysis or judge analysis output. The overall process of analysis may be iterative, but the individual analysis executions are not.

Work in (Horkoff, 2006) makes claims concerning the relationship between evaluation and model improvement, but does not relate these effects directly to the interactive nature of the procedure. As most existing work has not focused on the role of interaction in improving model accuracy or completeness, it is difficult to estimate how effective any technique may be in this area.

3.3.3 Domain Knowledge

R6 Prompt Further Elicitation: *The framework should contain analysis methods which reveals gaps in domain knowledge and prompts users to fill those gaps through further elicitation.*

Any analysis over a goal model can improve domain knowledge by providing some answer to a domain-related question. However, this knowledge only relates to what has already been captured in the model, assuming completeness and accuracy. In our requirement, we are interested in whether or not the procedures can reveal gaps in knowledge, prompting further elicitation and improved domain knowledge. Although it is certainly possible that this may occur when using any analysis procedure, most existing procedures do not address this requirement directly. Some procedures, such as the model checking procedures described in Section 2.3.5, or the forward satisfaction procedure of Letier & van Lamsweerde (2004), describe use of their procedures as a means to find problems in the specification represented by the model. Finding such errors could result in further elicitation, although there are no explicit studies which explore the nature or frequency of this elicitation.

3.3.4 Model Interpretation

R7 Definition: *The framework should contain a more formal or precise definition of the underlying agent-goal model framework. The analysis procedures should be formally defined in an effort to avoid divergent application or interpretation.*

All procedures contain an implicit interpretation of model constructs when describing reasoning over model syntax. Some approaches express model syntax explicitly by providing a formal definition for the underlying model and propagation (e.g., Giorgini et al., 2002; Letier & van Lamsweerde, 2004), while others describe the model and analysis procedure using more informal prose (e.g., Horkoff, 2006; Maiden et al., 2007).

3.3.5 Model Flexibility

R8 Accommodate Inexpressiveness: *The framework should contain analysis procedures which do not require formal or quantitative definitions of key model concepts.*

Many of the reviewed approaches require that the model have a precise formal or quantitative definition before undergoing analysis (e.g., Bryl et al., 2009b; Fuxman et al., 2003; Letier & van Lamsweerde, 2004), restricting application to very early stages of the requirements process. Approaches in (Barone et al., 2011; van Lamsweerde, 2009; Pourshahid et al., 2011; van Lamsweerde, 2009) do not require a formal model, but require the collection of various quantitative metrics or gauges based on real-life phenomena. Collection of such metrics in early requirements can be challenging, especially if metrics are required to be complete, i.e. at least one for each leaf goal. The approach described by (Barone et al., 2011) does account for incomplete available metrics (indicators), the implications of this approach to the current framework are discussed further in Section 13.3.

Other procedures use a quantitative interpretation over these informal concepts, assuming that the numbers are meaningful, i.e. customer satisfaction = 0.7 means that this goal is satisfied on a scale of 7/10 or has a 70% chance of being satisfied (e.g., Amyot et al., 2010; Giorgini, Mylopoulos, Nicchiarelli, & Sebastiani, 2004; Kaiya et al., 2002). This approach does not force users to give precise metrics for the model, but it does require quantitative estimation of initial values. The reliability of such quantitative results will be further assessed in Chapter 4

Several of the procedures address this requirement by supporting reasoning over flexible, inexpressive models via the use of simple, qualitative labels .

R9 Accommodate High-Level Domain Information: *The framework should contain analysis procedures which do not require detailed domain information, difficult to acquire in early RE stages.*

Several approaches require the addition of specific information such as cost, timing, or probability of occurrence in order to evaluate a model. Table 38 summarizes the extra information required for each procedure.

We differentiate between R8 and R9 by distinguishing between the supported expressiveness of the model and the information required by the domain. By accommodating inexpressiveness, we require that analysis procedures support inexpressive concepts, such as contributions or softgoals, without requiring an expressive definition, be it formal or quantitative. By accommodating high-level domain information, we require that analysis procedures do not require users to gather very specific or detailed domain information.

3.3.6 Decision Rationale

R10 Human Judgments: *The framework should support ways to capture, store and analyze the analysis decisions made over contentious areas in the model.*

We have listed some of the procedures that currently make use of human judgment, as it is defined in the above, in the consideration of R5 Interaction. These procedures ask for judgments to decide over partial or conflicting information (Chung et al., 2000; Horkoff, 2006) or to rewrite satisfaction arguments (Maiden et al., 2007). These judgments are associated with a particular part of the model, typically deciding the resulting value for one softgoal or goal. Other interactive procedures make judgments over the overall inputs or output of the procedures, focusing less on contentious areas in the model and more on deciding whether analysis output is satisfactory.

R11 Decision Rationale: *The framework should support ways to capture the rationale for decisions amongst alternatives, including varying analysis results which lead to these decisions.*

Although some procedures capture judgments over contentious areas of the model, few procedures capture the rationale for these judgments, or for decisions amongst analysis alternatives. The NFR Framework records "design rationale" by using a first class construct of a "claim" added to models. Claims can contribute to links just as goals contribute to each other. Claims and their contributions are included as part of the qualitative analysis procedure. Although such claims can provide helpful additional information, and can help to justify decisions between individual alternatives, they are limited in that they only apply to a particular link. They do not capture the rationale for decisions made over many factors. Also, there is an argument against adding rationale directly to models, as it has the potential to clutter models which are already complex, making analysis results even more difficult to understand.

Work in (Maiden et al., 2007) captures rationale for the satisfaction of means-ends targets using the notion of satisfaction arguments. However, these arguments are only applied to some areas of the model, and the approach does not account for partial satisfaction or negative satisfaction (denial). Kaiya et al. (2002) take a different approach to capturing rationale, attaching it to nodes and edges. This text explains why a modeler has decomposed a goal in a particular way, or why certain numeric contribution values are given. Although this text can be useful for justifying the structure of models, it does not directly help to rationalize localized or global analysis decisions. The AGORA approach also collects preferences attributed to individuals which could be used as a form of rationale for decisions. However, if these preferences are complex or conflicting, a further level of rationale may be needed to justify the final decisions.

3.3.7 Stakeholder Involvement

R12 Iterative Methodology: *The framework should contain clear methodologies to guide the process of interactive analysis including iteration over model content.*

Some of the techniques for goal model analysis reviewed in this Chapter provide only an analysis algorithm and do not provide an explicit methodology for use of the algorithm in practice (e.g., Amyot et al., 2010; Giorgini et al., 2002). Other approaches describe how their particular type of analysis can be used as part of a larger methodology. For example, Giorgini, Mylopoulos, & Sebastiani (2005) describe how the analysis procedures described in their previous work can be used as part of the early and late Requirements Analysis phases of the Tropos system development methodology. Liu, Yu, & Mylopoulos (2003) describe how i* analysis can be used as part of a methodology for identifying security and privacy requirements.

Others focus on technical aspects concerning how to apply the analysis procedure, but do not guide iteration over the model and analysis results, i.e. what if the results for existing alternatives are not good enough? (e.g., Chung et al., 2000; Horkoff, 2006; Letier & van Lamsweerde, 2004). The approach in (Bryl et al., 2006a) considers the possibility that the solution produced by the procedure may not be satisfactory. In this case, the recommendation is to iterate over other alternatives in the model, and not to modify or check the accuracy or completeness of the model itself.

As mentioned in our consideration of R4: Model Iteration, some methods include a brief description of model iteration as part of their approach. The planning method in (Bryl et al., 2006a) iterates until a suitable plan in terms of cost is found. If no satisfactory solution can be obtained, the approach recommends iterating over information in the model, or over the evaluation criteria. Work in (Liaskos et al., 2011) describes iteration over and improvement of the model and prioritization information as part of their planning method. Work in (Wang & Lespérance, 2001) recommends iteration over the i* model and its corresponding ConGolog model based on the results of a simulation. If errors or inadequacies are found in either model, the model will be changed, and the evaluation process repeated. Similarly, the model checking approach in (Fuxman et al., 2003) also advocates iteration over the model, based on the results of property checks. When checking properties and assertions over the Formal Tropos representation of an i* model, if the results were considered a “bug”, the Formal Tropos, and sometimes the i*, representation would be changed. Although consideration of model iteration as part of the methodology addresses our R11 requirement, the approaches do not provide explicit iterative methodologies, only acknowledging the possibility of iteration as a result of analysis.

As we reviewed in Section 2.4, there are several approaches aiming to guide construction of certain types of goal models. Although the focus is on model creation, some of these approaches include model analysis as part of their methodologies (Grau & Franch, 2007; Grau et al., 2005b; 2008). The approach in (Grau et al., 2005b) analyzes models at the end of the process by checking for consistency between i* models. Work in (Grau & Franch, 2007) introduces a process using Goal Question Metrics to evaluate alternative architectures in goal models. In (Grau et al., 2008), the authors again provide a detailed methodology, this one aimed to reengineer processes. After comparing metric results in order to find the best model alternative, they recommend iteration over the metrics used. In each of these procedures once the analysis stage is reached, it is assumed that the model is correct and sufficiently complete. Although these procedures are useful to guide model construction and the integration of various types of analysis in to the model construction process, they do not emphasize iteration over the model itself, or interaction with stakeholders.

3.3.8 Analysis Power

R13 Analysis Questions: *The framework should support a variety of analysis questions over agent-goal models, including "What if?" analysis.*

The goal model procedures summarized in this Chapter aim to answer a variety of questions such as "What is the effect of this alternative?", "How secure is the system represented by the model?", "What actions must be taken to satisfy goals?", "What happens when a particular alternative is selected?", "Is it possible to achieve a particular goal?". Although the procedures together cover a wide range of analysis questions, each individual approach can typically only answer one or two analysis questions. However, as can be seen in the goal model in Figure 5 there is a tradeoff between analysis power and procedure simplicity. Instead of creating one technique which combines many analysis approaches together to provide a high degree of analysis power, it may be more beneficial to apply selected approaches individually as a set of heuristics, aiming to keep the overall approach to model analysis simpler. The use of goal model analysis procedures as a heuristic is explored in more detail in Chapter 4.

R14 Reliable Analysis: *The framework should produce results which are accurate, sensible and reliable.*

It is difficult to assess the accuracy, reliability or sensibility of analysis results produced by existing work. We can only refer to the approaches which test their analysis method against industrial or illustrative examples with good success. Several methods describe such case studies, as listed in our consideration of R1 Scalability. Van Lamsweerde (2009) argues that existing qualitative analysis over goal models, as represented by the NFR framework, produce results which may be inaccurate, arguing for the use of procedures which draw on real measures from the domain. Although we agree that the accuracy of qualitative propagation over goal models is questionable, drawing real measure from the domain in early stages of analysis may not be possible or practical. We return to the issue of reliable analysis in Chapter 4 where we compare results using forward satisfaction analysis procedures.

3.3.9 Procedure Usability

R15 Simple Analysis Procedures: *The framework should contain analysis procedures which are simple enough to be applied with minimal training.*

The majority of goal model procedures reviewed in this work do not focus on the simplicity of their procedures. We can observe that several of the procedures seem quite complex with multiple stages and many assumptions made concerning the abilities of the user. For example, the approach introduced in (Asnar et al., 2007) combines techniques for satisfaction propagation, metrics analysis and planning, together with designer intervention in one approach. Although the method manages to provide much analysis power, by doing so it becomes very complex with many stages and branches of actions.

The quantitative, forward analysis procedure introduced by van Lamsweerde (2009) aims specifically to introduce lightweight analysis for goal models. However, this procedure combines together leaf goal prioritization, quantitative mappings of alternatives to leaves, with the cumulative propagation of gauge variables. Unless this approach is carefully implemented, perhaps in successive stages of information gathering and conceptual complexity, it may be difficult to apply in a participatory setting.

It is difficult to make an unbiased assessment of the simplicity or practical usability of any procedure without some form of empirical evaluation, such as studies with target users. Although some approaches use realistic case studies to validate the usability of their work, the focus of such studies is not on simplicity from the point of view of stakeholders, with model analysis likely performed by researchers.

R16 Tool Support Hides Complexity: *The implementation of analysis procedures in the framework should encode and hide as much complexity as possible from the user.*

Many of the goal model analysis approaches provide some sort of tool support. Most of these tools will hide much of the underlying complexity of the algorithms. For example, the GR-Tool described in (Giorgini et al., 2005) only requires the user to input analysis values, with the results calculated fully automatically. However, it is difficult to tell from the description of the implementations whether the tools hide enough of the complexity in order to be practically usable. Or whether the complexity is inherent to the nature of the procedure (for example, users have to be able to form questions and input initial values) and is not easily simplified via tool support. Similar to the assessment of R13, this is an area where user studies would help to demonstrate whether or not implementation results in a sufficient reduction in procedure complexity.

3.3.10 Summary

As we review existing approaches to goal model analysis and construction in light of our requirements for early RE, we see that although many approaches satisfy several requirements, none of them satisfies them all sufficiently. We return to our goal model from the previous chapter, summarizing our early RE analysis requirements. We place analysis values on the leaf requirements in this model based on our reflections concerning the ability of existing work, taken as whole, to satisfy (or deny) these requirements. Although these judgments can be subjective, we have justified our opinions with the evidence and examples in this section.

Figure 6 and Figure 7 show the forward evaluation results using these labels on the model without and with conflicts and synergies, respectively. We can see that in either case, many of the challenges in early RE analysis with agent-goal models are either only partially addressed, have a conflict value, are denied, or are unknown. This shows that although users could use a variety of approaches together in order to try and satisfy all requirements, they may still not have adequate results for handling model complexity, increasing model completeness or accuracy, or capturing decisions and rationale. The remainder of this thesis aims to develop an analysis framework for early RE which better addresses these challenges.

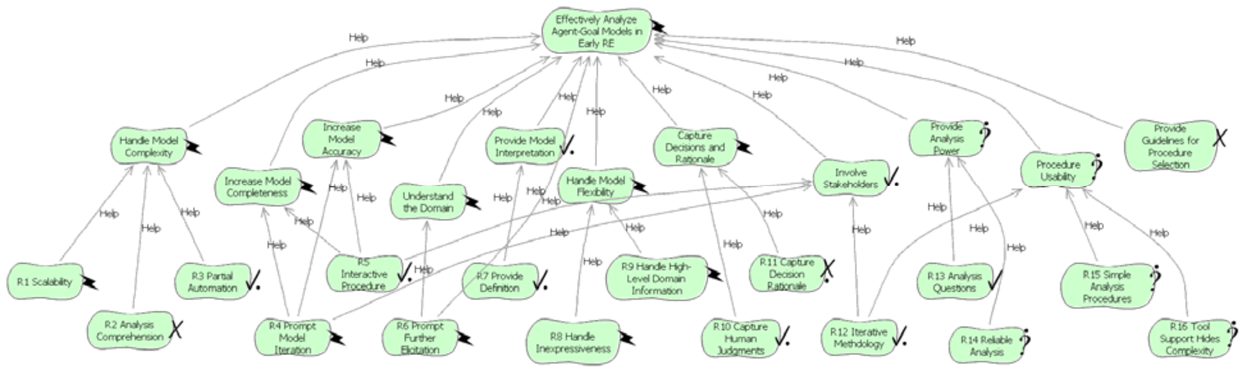


Figure 6: Satisfaction Analysis for the Requirements for analysis of Agent-Goal Models in Early RE based on a Combination of Existing Work

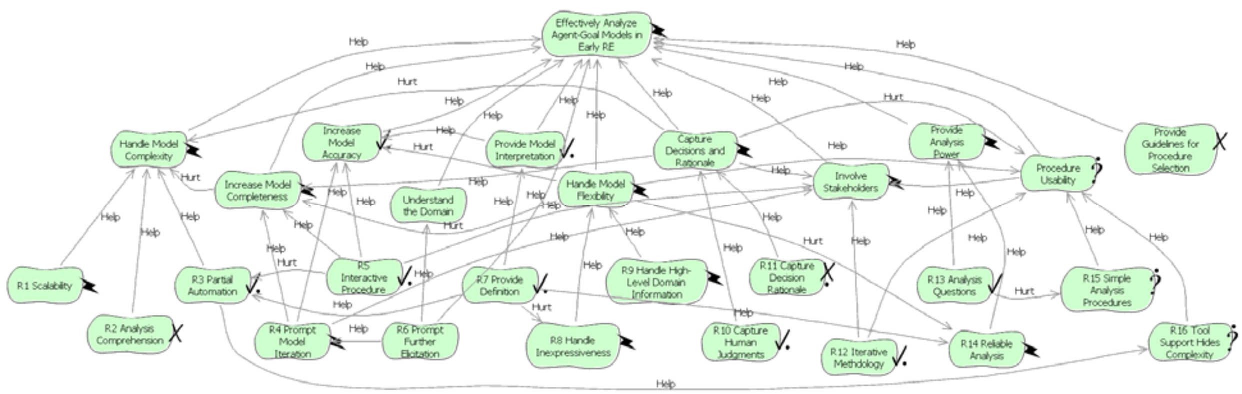


Figure 7 Satisfaction Analysis for the Requirements for analysis of Agent-Goal Models in Early RE based on a Combination of Existing Work including Conflicts and Synergies

3.4 Inclusion and Adaptation of Existing Agent-Goal Model Analysis Procedures

In developing a framework to address the requirements for early RE analysis with goal models, we could select any existing procedure to include and adapt as part of our framework. We make implicit use of our summary model above to consider the effectiveness of each procedure, also considering opportunities for improvement. The review in this section indicates that few procedures pay special attention to improving model quality through improved completeness and accuracy. Also, few procedures explicitly aim to involve stakeholders in the analysis process or to provide a simple and usable procedure. Only a subset of existing work supports model flexibility and avoids requirements for detailed domain information, handling the expressiveness needed to support early RE concepts.

We can also use our general guidelines described in Section 2.7 to help select procedures for use in early RE analysis (refer to Table 40 for specific questions). In such analysis, the domain contains a high degree of social interaction with many stakeholders (QU1), but we do not need to understand details or have access to specific domain information (QU2). We do need to communicate with stakeholders (QC1) and we are not confident in the accuracy and structure of our models (QM1). We are not yet ready to verify critical properties over our model (QM2), but we would like to determine system scope (QS1). We need to find high-level requirements, but not yet detailed requirements (QE1, QE2). We would like to consider non-functional requirements and capture domain assumptions (QE3, QE4). It is not known whether we are working in a system where safety, security or privacy is critical, and our requirements are not yet specific enough to search for errors or inconsistencies automatically (QR1, QR2). Likely, we are looking for more high-level design alternatives (QD1), but do not yet need specific design alternatives (QD2). We need to evaluate high-level and not detailed alternatives (QD3, QD4). It is likely that very early analysis is not yet interested in process, and does not have enough detailed information to test run-time operations (QD5, QD6).

Our guidelines recommend the use of the forward qualitative analysis procedures in (Maiden et al., 2007) and (Horkoff, 2006). Although the two approaches bear similarities, we choose not to include and expand upon the procedure in (Maiden et al., 2007) for a number of reasons. The nature of propagation in this approach is limited to only compliance or non-compliance, not allowing degrees of satisfaction or denial. The procedure propagates compliance originating from only one requirement/task at a time, as opposed to a set of selected initial values, allowing representation of complex analysis procedures. The approach is designed to test the compliance of a set of existing requirements, whereas in early RE, such requirements are likely to not yet exist. Although the use of satisfaction arguments in this approach is novel, and meets one of our early RE Analysis requirements (R11), it is limited to only a subset of model structures (goals, softgoals), and does not support rationale for non-satisfaction (denial).

Other procedures with strong recommendations, according to our guidelines, include metrics evaluation of (Franch, 2006; 2009), the combination procedure in (Asnar et al., 2007), and the simulation procedure in (Gans et al., 2005). However, these approaches may be too complex to encourage communication with stakeholders.

Using these considerations, we select the procedure in (Horkoff, 2006) to use as a basis for the work in this thesis. We summarize the benefits and drawbacks of this work in an analysis of our summary model in Figure 8. We can see that although the overall judgment for effective analysis of agent-goal models in early RE is still a conflict value; the procedure partially addresses improving model quality, involving stakeholders, and model flexibility. The procedure offers only the forward type of "what if?" analysis questions. Human judgments are used in analysis, but not stored for later viewing. The procedure has provided some evidence for the scalability and the simplicity of the procedure, applying the procedure manually to large models; however the procedure was only applied by the author. The work does not address analysis comprehension, does not provide a precise definition of concepts, and does not provide a methodology. Tool support was made available, but was not extensively used.

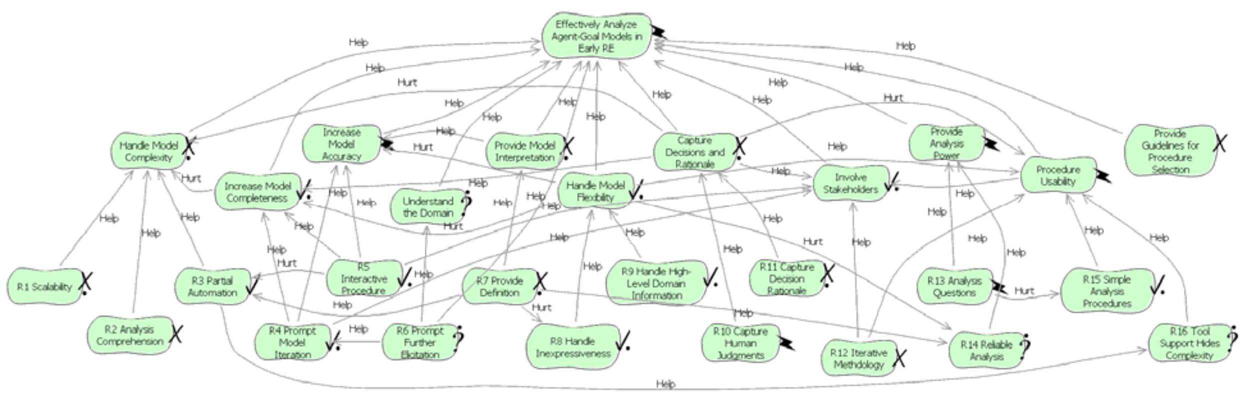


Figure 8: Satisfaction Analysis for the Requirements for analysis of Agent-Goal Models in Early RE based the contributions of Horkoff (2006)

Work in (Horkoff, 2006) has used the i* syntax as an example agent-goal model framework. When selecting an agent-goal model syntax to use in our framework, several options are available, including GRL, Tropos, KAOS, and i* (Amyot, 2003; Bresciani et al., 2004; Dardenne et al., 1993; Yu, 1997), as summarized in Section 2.1. The i* Framework has been used as a basis for agent-goal modeling in the GRL and Tropos Frameworks, with GRL simplifying i* actor (agent) syntax and Tropos expanding the framework to include an agent-oriented software development methodology. KAOS does not support informally or imprecisely defined softgoals, all goals must have a clear decomposition or have their achievement specified probabilistically (Letier & van Lamsweerde, 2004), making it more suitable for later RE specification and analysis. In this work, we follow (Horkoff, 2006) by using the i* Framework to ground our examples and procedures in a specific syntax. Procedure and methods introduced in this work could be easily applied to other goal modeling frameworks, such as GRL, Tropos, and the NFR Framework, and could be applied with more effort to Frameworks such as KAOS, AGORA, or GBRAM.

The remainder of this work builds upon the analysis procedure in Horkoff (2006), creating a framework for early RE analysis which provides more analysis power, better handles model complexity, provides a more formal interpretation of the model, and employs extensive empirical evaluation to test the claims of the framework as a whole.

3.5 Conclusions

In this chapter, we have justified our selection of agent-goal models for early RE analysis. Challenges of early RE analysis gathered from related work and listed in Chapter 1 have been used to compose a list of requirements for early RE agent-goal model analysis. We have evaluated existing goal model analysis procedures in light of these requirements. We have used this evaluation, as well as the general guidelines provided in the previous chapter to select procedures to be included in and expanded upon in the framework produced in this thesis.

In the next Chapter, we continue our comparison and evaluation of goal-oriented analysis procedures, this time focusing specifically on the Reliable Analysis (R13) requirement for forward satisfaction propagation techniques.

Chapter 4 Forward Satisfaction Analysis Techniques for Goal models – Detailed Comparison

In Chapter 3, we have gathered requirements for early RE agent-goal model analysis, using these requirements to evaluate the suitability of existing procedures for use in early RE. Although we were able to assess how well requirements such as scalability (R1) and model iteration (R4) are addressed by existing work, we were unable to judge how well existing procedures satisfy R14, reliable analysis.

As can be seen in our Chapter 2 review, existing work in goal model analysis emphasizes the analytical power of goal model analysis procedures. Much of such work focuses on the conclusions which can be drawn from the models, emphasizing their role as a decision making tool, helping modelers to choose between alternative system functionality or design configurations. However, in Section 2.7, we have listed benefits of goal model analysis beyond analytical power both for early RE and general system development stages. For example, iterative and interactive analysis could be used to improve the quality of the model or the understanding of the domain by forcing examination of sections of the model, or by checking the contents of the model against user understanding. Careful consideration of the model prompted by analysis or consideration of the analysis results themselves can lead to further requirements elicitation, filling gaps in knowledge. Model analysis can be used as a means of communication between and amongst stakeholders and analysts concerning the effects of alternatives or properties of the model, aiming for convergent understanding of the domain. By performing comparisons to check the reliability of analysis results, we evaluate whether or not certain goal model analysis procedures are best used as a decision making tool, or are better used to achieve other benefits, as listed.

This chapter is an expansion of the following papers/reports:

Horkoff, J., & Yu, E. (2011b). Comparison and Evaluation of Goal-Oriented Satisfaction Analysis Techniques. Requirements Engineering (REJ) (conditionally accepted).

It is difficult to judge the accuracy of analysis performed over high-level, social models capturing the "to-be" space. However, we can begin to judge the reliability of analysis results by comparing results across similar procedures. If results are reliable, similar analysis approaches should produce very similar results over the same models. Several procedures reviewed in Chapter 2 analyze the satisfaction or denial of goals (Section 2.3.1). These approaches differ in several dimensions, including the specifics of propagation through links, interpretation of goal model syntax, measurement choices for goal satisfaction, and the level of participation of the user. The forward evaluation procedure described in this framework makes a set of procedural and interpretation choices similar to (Horkoff, 2006). It is unclear how these different interpretations and choices would affect analysis results. In this work, we aim to understand the practical consequences of these different procedural choices, how they reflect on the reliability of procedure analysis results, and how they would affect use of evaluation in practice.

In this Chapter, we focus on comparing and analyzing the differences amongst procedures which propagate satisfaction values forward through model links. To make the comparison, we use examples of goal models from the literature, and apply a selection of available procedures to analyze several alternatives within the example models. We define conventions for comparing differing result formats. Variations in the results are analyzed, including the design alternative each procedure appears to favor. The purpose of the analysis is to understand to what degree variants in procedure design affects analysis results. We use these results in part to evaluate the R14, reliable analysis, requirement, and in part to understand potential benefits of goal model analysis, including how goal model analysis could be used effectively in practice.

4.1 Procedure Selection for Comparison

As reviewed in Section 2.3.1, goal satisfaction procedures start with initial values assigned to the model, reflecting an alternative or question, and then use model links to propagate values either forward (in the direction of the link), or backward. These procedures can answer questions like "What is the effect of this alternative?" (forward) or "Can these goals be satisfied?" (backward). See Table 17 for a summary of satisfaction propagation procedures.

When defining propagation over goal models, satisfaction analysis techniques make different interpretations of certain goal model concepts and make differing procedural choices. Some satisfaction analysis procedures present results in terms of qualitative labels representing satisfaction or denial, typically using: (sufficiently) satisfied, partially satisfied, (sometimes) conflict, none/unknown, partially denied, and denied. For example, the initial definition of the softgoal concept in the NFR Framework avoided a precise definition in order to allow for user judgment and flexibility (Chung et al., 2000). Several procedures offer quantitative analysis, using numbers to represent the probability of a goal being satisfied or denied, or to represent the degree of satisfaction/denial (Amyot et al., 2010; Giorgini et al., 2005). Other procedures produce only binary results, where goals have only one of two values, typically satisfied or not (Maiden et al., 2007). Some procedures apply a more formal definition to the goal model concepts, using predicate logic or algorithms to determine their satisfaction levels automatically (Giorgini et al., 2005; Letier & van Lamsweerde, 2004).

One of the primary distinguishing features between these approaches is their means of resolving multiple incoming values for goals. Goal models often include contribution links representing positive and negative consequences of various degrees (see Figure 1 and Figure 2 for examples). A goal could receive several different types of contributions at once, positive and/or negative of various strengths. Some procedures deal with such situations by separating negative and positive evidence, making it unnecessary to resolve conflicts (Giorgini et al., 2005). Other procedures make use of predefined qualitative or quantitative rules to combine multiple values (Amyot et al., 2010). Further procedures are "interactive", using human intervention based on domain knowledge to resolve partial or conflicting evidence (Chung et al., 2000; Horkoff, 2006).

For the purpose of the comparison in this Chapter, we select a subset of goal model analysis techniques for a more detailed evaluation and comparison, specifically, qualitative and quantitative satisfaction analysis techniques which propagate satisfaction levels in the forward direction. Specifically, we select seven procedures over three alternatives in three models using three tools. We select the following procedures: the three GRL procedures (quantitative (GRL-quant), qualitative (GRL-quant), and hybrid (GRL-hybrid)) (Amyot et al., 2010); the interactive, qualitative procedure introduced as part of the NFR Framework (NFR) (Chung et al., 2000); the forward qualitative and quantitative procedures associated with the Tropos methodology

(Tropos-qual), (Tropos-quant) (Giorgini et al., 2005); and the qualitative procedure aimed for forward analysis of i* models (i*) included in this thesis. Each of the first six procedures has been summarized in Section 2.3.1. The forward analysis procedure which is part of the framework introduced in this thesis is an expansion and adaptation of the procedure used in (Horkoff, 2006), which was itself an expansion of (Chung et al., 2000). Details concerning this expansion, including a more precise definition of propagation can be found in Chapter 5.

We omit other procedures which propagate forward satisfaction values, such as (Maiden et al., 2007), analysis in AGORA (Tanabe et al., 2008), quantitative analysis over goal trees (Lamsweerde, 2009), and analysis in KAOS (Letier & van Lamsweerde, 2004), as these procedures are too dissimilar to the selected seven to produce a clear comparison. Specifically, the approach in (Maiden et al., 2007) differs in the scale of measure and the coverage of propagation. The approach only propagates compliance or non-compliance and not degrees of satisfaction or denial. The procedure propagates compliance originating from only one requirement/task at a time, as opposed to a set of selected initial values as is done in the seven selected procedures. The impact analysis procedure described in (Tanabe et al., 2008) for AGORA is focused on change management, detecting conflicts when a new goal is added and analyzing goal achievement when a goal is deleted. When a goal is added, the procedure uses goal characteristics such as security or usability to suggest conflicts between goals. When a goal is deleted the approach calculates impact on the parent goal using a ratio of the contribution values assigned to the links. Unlike in the selected procedures, this value is not propagated further up the graph. The AGORA procedure can also calculate achieve and obstruct values for the roots goals in the graph. As this type of propagation is very similar to the Tropos quantitative evaluation (Giorgini et al., 2005), we omit it from our comparison.

The approach in (Lamsweerde, 2009) relies on the presence of quantitative gauges collected from the domain, such as time, cost, quantities, etc. As the other procedures under comparison do not require or specify how to deal with such concrete measures, comparison between this approach and others is difficult. Furthermore, the example models we select from existing procedures (Section 4.2) do not come with the gauges required by this procedure. Future work could find examples where such gauges are available and compare results between analysis procedures directly and indirectly using such measures.

Work in (Letier & van Lamsweerde, 2004) produces degrees of probabilistic satisfaction, but requires additional, specific information in the form of cumulative distribution functions over random variables for goals in the model. The sample models available for each of the other procedures does not contain this information and the information would not be explicitly used by any of the procedures, making a comparison difficult.

We also omit the backward propagation procedure (Giorgini et al., 2004b) as the form of analysis question between backward and forward propagation is different ("what if?" vs. "is this possible?") making the results difficult to compare.

Several of the techniques compared in this work have been expanded to allow further analysis capabilities. The backwards approach in (Giorgini et al., 2005) allows for the addition of analysis constraints, conflict restrictions, and finding a minimum cost solution. (Asnar & Giorgini, 2006) expands on (Giorgini et al., 2005) to include quantitative analysis of acceptable risk levels and costs. This procedure works over an expansion of the Tropos Framework which includes events, risks, and (risk) treatments. Wang, McIlraith, Yu, & Mylopoulos (2007) adapt the work in (Giorgini et al., 2004b), using goal models to diagnose run-time failures. (Amyot et al., 2010) uses quantitative, qualitative or hybrid analysis and use per-actor goal priorities added to the models, to calculate an overall numeric satisfaction value for an actor. We do not consider these extended features in this study.

4.1.1 Selected Satisfaction Analysis Techniques: Objectives and Methods

When defining propagation over goal models, our selected satisfaction analysis techniques make different interpretations of certain concepts and their relationships. These differences can be attributed to different assumptions concerning the use of goal models in practice, including the objectives of goal model application and how goal models would be used as part of a system development methodology.

For example, the NFR Framework (Chung, Nixon, E. Yu, & Mylopoulos, 2000) provided their analysis technique as a means to determine the impact of design decisions on high-level softgoals. Analysis is intended to be applied after iterative stages of elicitation, NFR identification, operationalization, and decision making. The initial definition of the softgoal concept avoided a formal or quantitative definition, in order to allow for user judgment and flexibility in dealing with non-functional requirements.

Similarly, the approach described in (Horkoff, 2006) leaves softgoal resolution to the user in order to allow for an interactive process which compensates for the incompleteness of models in the early RE process. Evaluation is again performed after an iterative process of elicitation and modeling. However, this approach aims to help analysts make decisions over alternatives in the model, as opposed to evaluating decisions currently made. The work encourages modelers to add knowledge gained as part of the evaluation process to the models, in order to improve their quality.

Other procedures (e.g., Amyot et al., 2010; Giorgini et al., 2005) apply a more formal definition to the softgoal concept, either using predicate logic or algorithms to determine their satisfaction levels automatically. GRL evaluation (Amyot et al., 2010) acknowledges that goal models can be applied to achieve several purposes, such as assessing goal satisfaction, evaluating design alternatives, deciding on high-level requirements, testing model sanity, and supporting communication. Given the varying purposes for goal model analysis, this work attempts to support a variety of qualitative and/or quantitative analysis approaches. The description focuses on the evaluation algorithms themselves, and not how these algorithms fit into an overall modeling and system analysis process.

The Tropos methodology (Giorgini et al., 2005) aims to support all software development phases, including early and late requirements analysis as well as architectural and detailed design. Tropos goal model analysis is intended to answer questions such as "given the satisfaction of as set of leaf goals, can root goals be fulfilled?" and "which set of leaf goals (if any) fulfill all root goals?". Presumably, this type of analysis can be applied in both early and late requirements analysis, although the specific role of analysis in these phases is not described.

Overall, by observing the varying approaches to softgoal definition and resolution (interactive/automatic), we see that some approaches treat goal models as an exploratory tool, capturing imprecise and incomplete information, while some use them as more precise definition of system boundaries, intentions and interactions. The former assumption would assume that user intervention is needed to compensate for model imprecision and incompleteness, while the latter would assume that the model is fit for automated analysis.

Similarly, some procedures allow for the use of available quantitative measures (e.g., Amyot et al., 2010; Giorgini et al., 2005), while others avoid such measures (e.g., (Chung et al., 2000). These choices reflect an underlying assumption about the potential availability of accurate numerical domain information, which, in turn, reflect assumptions about the types of systems and domains under analysis (accurate metrics or user estimations readily available or not) or the stage of the project when goal model analysis is applied (early, exploratory stages or later requirements or design stages).

As goal models and goal model analysis procedures can be applied in to a variety of domains and can play a role in multiple stages of a project, we make no assumptions about the "right" way to interpret goal model concepts, or the "right" level of assumptions concerning available metrics. The purpose of this exercise is not to find the "best" technique, but to understand to what degree different assumptions about goal concepts and propagation effect procedure results. If the results do not vary significantly, then the choice between available procedures may not be significant, and we may assume that satisfaction propagation techniques provide a level of reliability in their analysis results (R14). If, however, results vary widely, then we note that the differing interpretations of goal model concepts are significant, and should be used to guide potential users in how and for what purposes they apply goal model analysis. Although the focus of the framework introduced in this thesis is early RE analysis, generally, we intend to use the analysis and comparison of results in this Chapter to help understand how and in what contexts existing procedures can best be used.

4.2 Selected Sample Models

As sample models, we select models used by the original authors to introduce the analysis procedures: the Media Shop model (Giorgini et al., 2005) (Figure 9), the Wireless Service model (Amyot et al., 2010) (Figure 10), and the Counseling Service model used to illustrate one of the case studies described in this work (Figure 11). The latter study has been summarized in Section 1.1.1 and will be described in more detail in Section 12.1.4. We select these models as they are of a sufficient level of complexity to facilitate interesting analysis results, but are large enough to produce results which may be overwhelming. The sizes of the models are in a similar range, 33, 16, and 31 elements for Figure 9, Figure 10, and Figure 11, respectively. Each of these models makes effective use of the goal model constructs used in each paper. Finally, the models provide three dissimilar domains over which to test the analysis procedures.

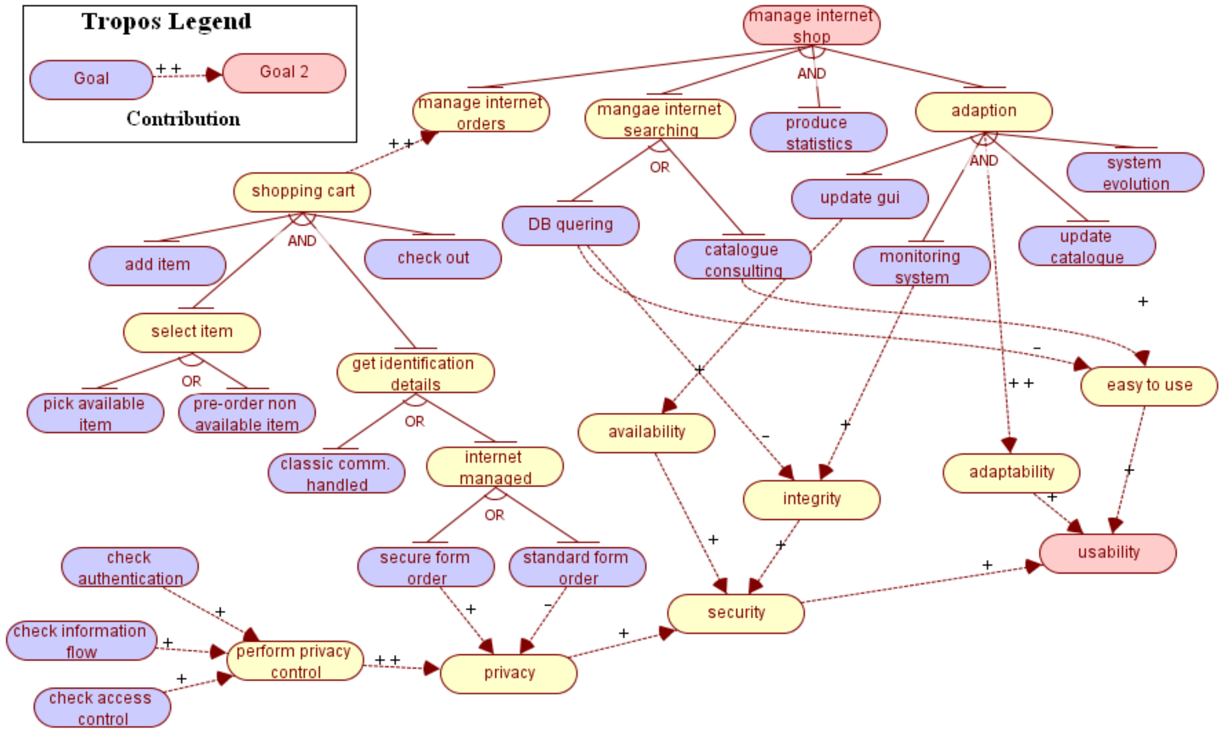


Figure 9: Tropos Actor diagram from the Media Shop example appearing originally in (Giorgini et al., 2005)

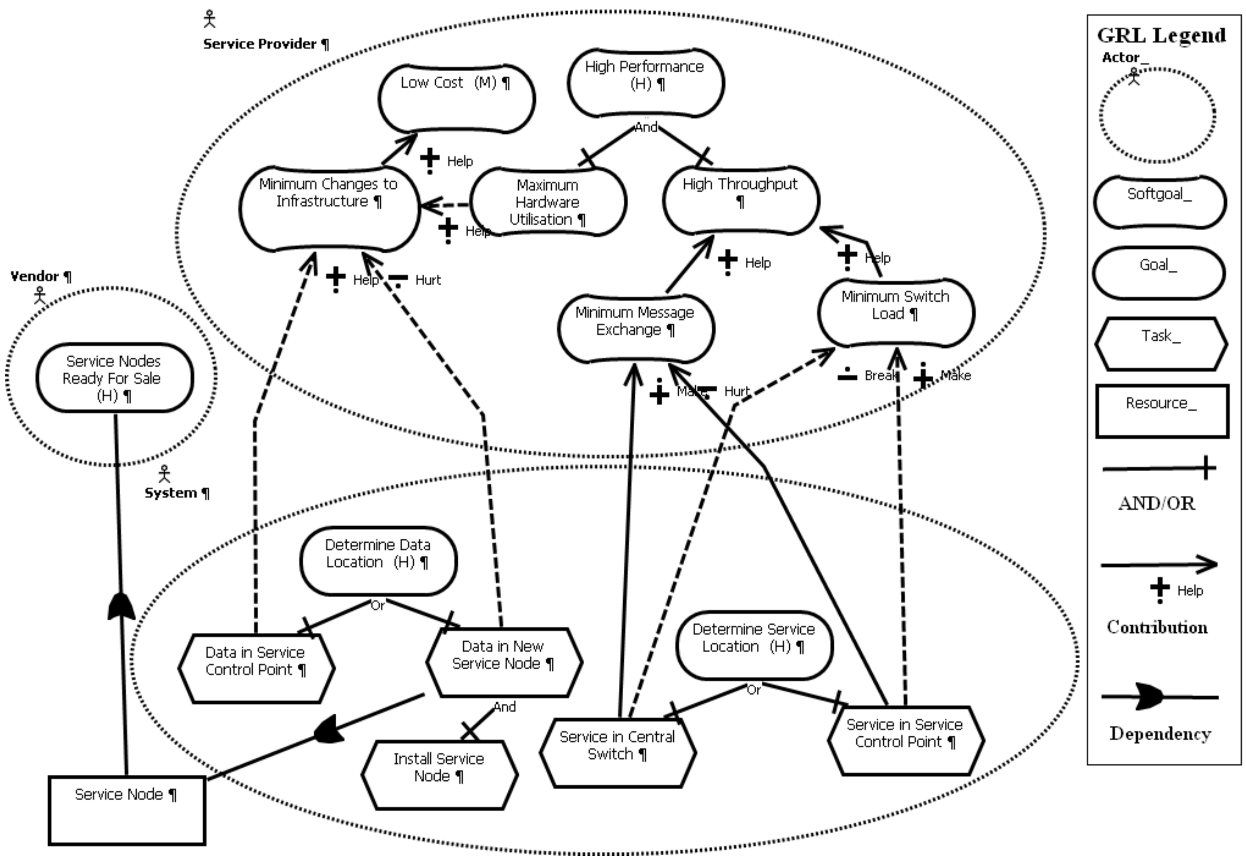


Figure 10: GRL model of a Wireless Service appearing originally in (Amyot et al., 2010)

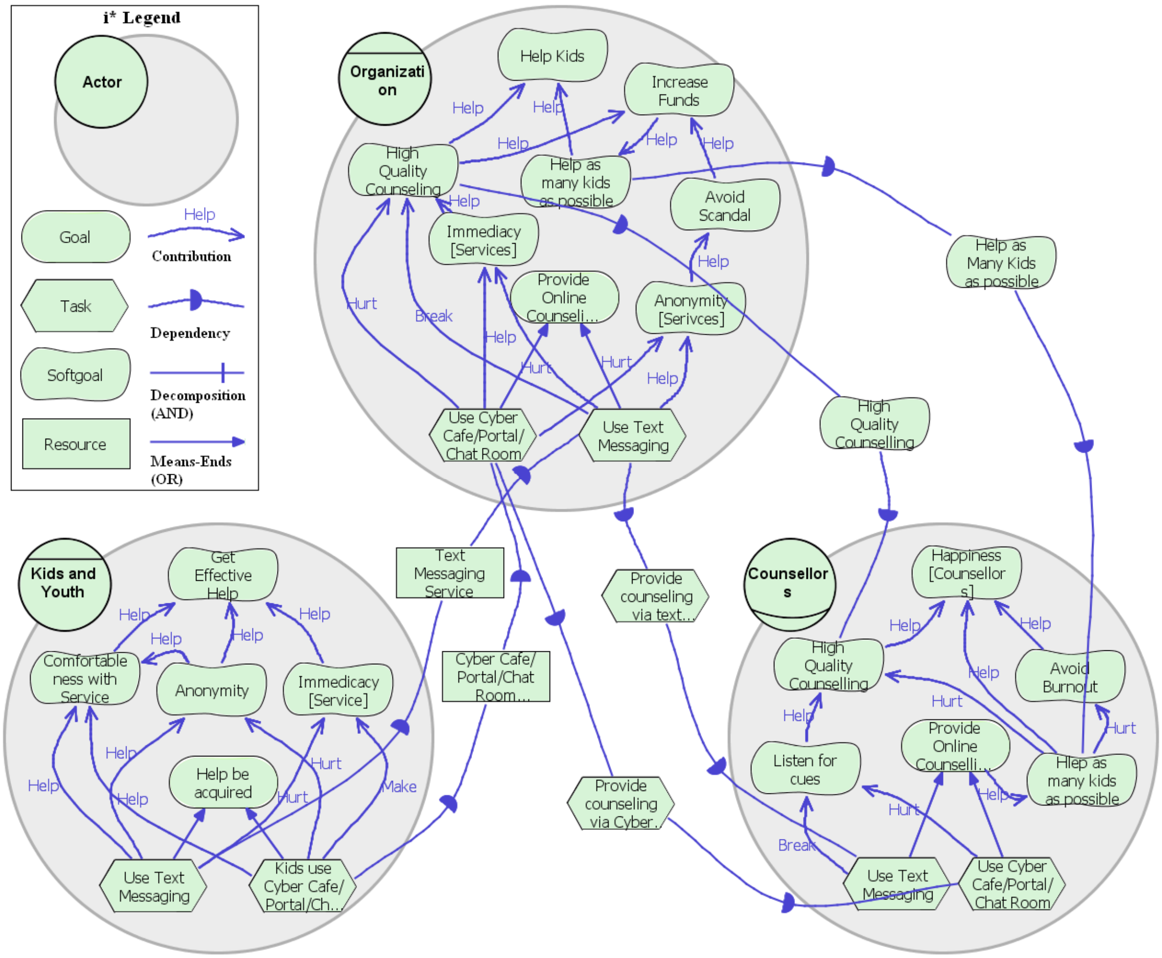


Figure 11: i* Counseling Service Model repeated from Figure 2

4.3 Selected Model Alternatives

Within each model, we select three alternatives most likely to produce the most diverse results. For the first two models, alternatives were taken from the original papers. For the Media Shop example in Figure 9, we select alternatives 1, 2, and 4 from Table 1 in (Giorgini et al., 2005). We modify these alternatives slightly by adding initial satisfied values to the leaf goals not involved in OR relationships, otherwise the results would not match what appears in (Giorgini et al., 2005). We select these alternatives as they have roughly the most dissimilar initial values, in order to produce a wider range of results. For the alternatives applied to Figure 10, the Wireless service example, we select alternatives 1, 5, and 6 from Table 7 in (Amyot et al., 2010) for the same reasons. As there are two alternative tasks in the Counseling Service model in Figure 11, we select each alternative individually (Use Text Messaging and Use Cyber Café/Portal/Chat Room), and the alternative where both tasks are satisfied, producing three alternatives. The initial values for all three alternatives over all three models are summarized in Table 41, Table 42, and Table 43.

Table 41: Initial evaluation values for three alternatives used with Figure 9 from (Giorgini et al., 2005)

Element	Alt 1	Alt 2	Alt 3 (4)
DB querying	100, FS	100, FS	
catalogue consulting			100, FS
pick available item	100, FS	100, FS	100, FS
classic communication handled	100, FS		
standard form order		100, FS	100, FS
monitoring system	100, FS	100, FS	100, FS
produce statistics	100, FS	100, FS	100, FS
system evolution	100, FS	100, FS	100, FS
add item	100, FS	100, FS	100, FS
check out	100, FS	100, FS	100, FS
update catalogue	100, FS	100, FS	100, FS
check authentication	100, FS	100, FS	100, FS
check information flow	100, FS	100, FS	100, FS
check access control	100, FS	100, FS	100, FS
update GUI	100, FS	100, FS	100, FS

Table 42: Initial evaluation values for three alternatives used with Figure 10 from (Amyot et al., 2010)

Actor	Element	Alt 1 (1)	Alt 2 (5)	Alt 3 (6)
Service Provider	Maximum Hardware Utilization	50, PS	50, PS	50, PS
	Data in Service Control Point	100, FS		
	Service in Central Switch	100, FS		
	Service in Service Control Point		100, FS	100, FS
System	Install Service Node		100, FS	100, FS
(dependum)	Service Node			100, FS
Vendor	Service Nodes Ready for Sale		-100, FD	100, FS

Table 43: Initial evaluation values for three alternatives used with Figure 11

Actor	Element	Alt 1 (1)	Alt 2	Alt 3
	Use Text messaging	100, FS	-100, FD	100, FS
Counselors	Use Cyber Café/Portal/Chat Room	-100, FD	100, FS	100, FS

4.4 Selected Tool Support

Each of the seven procedures has provided a tool implementation, with the exception of the NFR procedure. All of the tools are freely available for download. We make use of each of these tools to apply the procedures to our three example models, redrawing each of the three models in each tool. Specifically, we apply the evaluation techniques using the jUCMNav tool (JUCMNav, 2011) for the three GRL techniques, OpenOME ("OpenOME, an open-source requirements engineering tool," 2010) for the i* and NFR technique (with manual adjustments to human judgment criteria in the application of the NFR technique), and the GR Tool (GR-Tool, 2010) for the two Tropos techniques. The OpenOME tool is used to implement the tool supported components of the framework introduced in this work; more detail will be presented in Chapter 11. A summary of our overall comparison approach including sources, procedures, tools and models is shown in Figure 12.

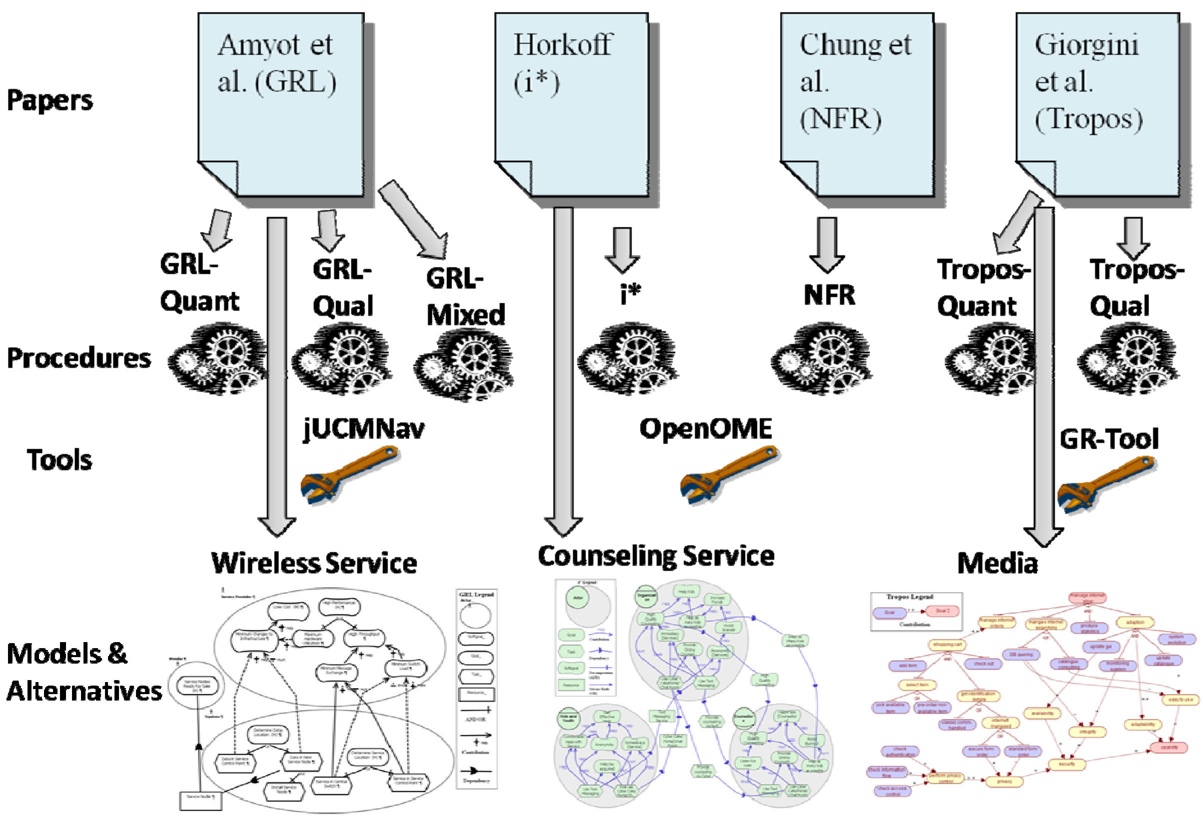


Figure 12 Summary of the Comparison Process for the Seven Procedures over Three Models using Three Tools

4.5 Conversions, Adjustments and Conventions

Although the procedures have much in common, some conversions, adjustments and conventions need to be made and adopted in order to allow for the results to be more easily compared. We endeavor to only convert formatting, without affecting the results themselves. Cases where adjustments may affect procedure results are changed deliberately in order to test the impact of such changes.

Measurement Values. The qualitative techniques use similar but slightly differing labels, for example PS (partially satisfied), ✓· (partially satisfied), and W^+ (weakly satisfied). For this comparison we will convert all values to a common scale of fully satisfied (FS), partially satisfied (PS), conflict (C), none (N), partially denied (PD), and fully denied (FD). The Tropos procedures produce two results, one for satisfied and one for denied (Sat, Den), while other procedures produce only a single value. We leave the two values as is, without introducing some form of combining automatic values, allowing the reader to make comparisons. The GRL-quant and –hybrid procedures use a scale from -100 to 100, while the Tropos-quant procedure uses 0 to 1.0 for both Sat and Den. We leave these values as is; however a comparison can be made by dividing the GRL result by 100 and moving it to the Sat (+) or Den (-) side (for example (-37 = 0, 0.37). When selecting initial values to start analysis we convert FS = +100 = Sat: 1.0 and FD = -100 = Den: 1.0.

When counting differences between qualitative and quantitative results, we use a rough translation of FS/FD = +/-95 to 100 (0.95 to 1.0), PS/PD = +/-5 to 94 (0.05 to 0.94). We treat N and C as different values, making the distinction between no evidence and conflicting evidence.

Human Judgment. Some techniques (i*, NFR) require human intervention to resolve evidence, we indicate these decisions by presenting the results in parenthesis, for example (PS). Whenever possible, the same judgments are used across all alternatives, in other words the evaluator does not change her mind from one alternative to the next. The judgments made are intended to be reasonable, reflecting the evidence presented by the model.

In applying the NFR procedure, we use the original description, where all final values must be promoted to one of FS, FD, C, or Unknown, and where Conflict or Unknown is selected whenever present in human judgment. The i* approach adopted in the forward evaluation procedure in this thesis relaxes these rules for human judgment, allowing the user to choose ignore conflicts or unknowns and decide on partial resulting values.

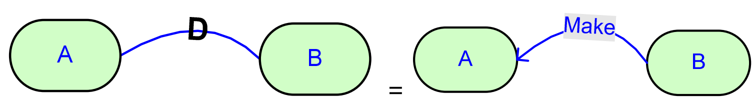
Cycles. Because the jUCMNav tool did not allow us to draw a two-goal loop, and to simplify the comparison, we remove one of the links in the original counseling model, making results potentially different from what will appear in Chapter 6.

Dependency Links. As the NFR and Tropos procedures do not explicitly support dependency links, we have treated these links to *make* (++) contribution links when using these procedures over models with dependency links, for example, if x depends on y, then y makes x.

Some+ and Some-. The NFR and Tropos procedures do not support the difference between help and some+ (hurt/some-) contribution links. Some+/Help and Some-/Hurt are treated identically in both the i* and the GRL-Qual procedure, only potentially effecting human judgment in i* analysis. In order to equalize the ability of the procedures, Some+/- links are treated as help and hurt links when undergoing these four evaluation procedures (NFR, Tropos, i*, and GRL-Qual). The GRL quantitative and hybrid procedures automatically convert Help and Some+ links to 25 and 75, respectively, although the quantitative algorithm allows the optional definition of link-specific numeric values. For the application of the GRL-Hybrid procedure, we have converted all Some+/- links to Help/Hurt, respectively, meaning they all have a value of +/-25. If we made the same conversion for GRL-Quant, results for the GRL-Quant and GRL-Hybrid would be very similar to each other, but not identical, due to algorithm differences (Amyot et al., 2010). However, in order to understand the significance of numeric label selection on procedure results, we have modified the models to give help/hurt links a value of +-50 in the GRL-quant algorithm, while the hybrid algorithm retains the original conversion. Quantitative evaluation result will now differ from what appears in (Amyot et al., 2010). As it is, there are only two Some+/- links in all of our sample models, both in the Wireless Service Model.

Link Symmetry. The Tropos procedures contain contribution links which can be asymmetric, propagating only positive (s) or only negative (d) evidence, while other approaches have symmetric links, propagating both positive and negative evidence. The model and results in (Giorgini et al., 2005) are intended to reflect symmetric links; however, a closer observation reveals that the results presented in the paper and the implementation of the G-R Tool only use asymmetric positive links, i.e., only positive evidence is propagated. We use this convention in our trials, partially as a means to determine its impact. Results in this work differ from results in (Giorgini et al., 2005) only in value of one goal, integrity, in Alternative 2. This goal has a value of partially negative in our results. We suspect this difference is due to an omission of a hurt link in the sample file available with the GR-Tool when compared to the model as shown in (Giorgini et al., 2005), i.e. the results reflect a model slightly different than the model shown in (Giorgini et al., 2005) and in Figure 9. Conversions, adjustments and conventions used between procedures are summarized in Table 44.

Table 44: Summary of Conversions and Conventions used between the seven Analysis Procedures

	GRL-Qual	i*	NFR	Tropos-Qual	Tropos-Quant	GRL-Quant
Analysis Results	✓ ✓• ⇝ ✗• ✗	✓ ✓• ⇝ ? ✗• ✗	√ W^+ U W^- X	Sat: S, PS, N Den: D, PD, N	Sat: 0 to 1.0 Den: 0 to 1.0	-100 to 100
Analysis Results Conversion	✓=√= Sat: S, ✓•=W^+=Sat: PS, ?=U, ✗•=W^-=Den: PD, ✗=X=Den: D, None=N				Sat_Tropos = +GRL/100 Den_Tropos = -GRL/100	
Conversion for Results Comparison	S = 95 to 100 (Sat: 0.95 to 1.0), PS = 5 to 94 (Sat: 0.05 to 0.94) D = -95 to -100 (Den: 0.95 to 1.0), PD = -5 to -94 (Den: 0.05 to 0.94) C != N					
Dependency Links	Supported	Supported	Not Supported	Not Supported	Not Supported	Supported
	 					
Some+ and Some- Links	Supported	Supported	Not Supported	Not Supported	Not Supported	Supported
	Some+ = Help, Some- = Hurt					
Contribution Link Symmetry	Symmetric	Symmetric	Symmetric	Asymmetric Positive	Asymmetric Positive	Symmetric
Human Judgment Initiation	Not Used	Conflicting or partial values	Conflicting or partial values, ♮or U	Not Used	Not Used	Not Used
Human Judgment Results	Not Used	Any value	No partial values	Not Used	Not Used	Not Used

4.6 Results

We provide tabular results for all three models in Table 45, Table 46, and Table 47. In these tables, we list the model elements (goals, softgoals, task, resources) on the left, including the actor whose boundary the element appears in. The next set of columns, Alternative 1, presents the results of all seven procedures for the first alternative. Alternatives are distinguished by initial values extracted from the papers, listed in Table 41, Table 42, and Table 43, and marked in the tables below with an asterisk (*). The next two sets of columns, Alternative 2 and 3, provide the same information for the second and third alternatives extracted from the paper. Values in parentheses are the result of human judgment. We use the conversions in Table 44 to identify results which differ. When the result from one procedure differs from the rest for one element per alternatives, we highlight this result using bold with a grey background. When there is a significant difference among the results for two or more procedures, we highlight the whole partial row for that element in that alternative.

Table 45: Results of the Evaluation Techniques Applied for Three Alternatives over the Media Shop Model in Figure 9

	Alternative 1							Alternative 2							Alternative 3						
Element	*GRL-Quant*	*GRL-Qual*	*GRL-Hybrid*	*i**	*NFR*	*Tropos-Qual*	*Tropos-Quant*	*GRL-Quant*	*GRL-Qual*	*GRL-Hybrid*	*i**	*NFR*	*Tropos-Qual*	*Tropos-Quant*	*GRL-Quant*	*GRL-Qual*	*GRL-Hybrid*	*i**	*NFR*	*Tropos-Qual*	*Tropos-Quant*
DB querying	*100	*FS	*100	*FS	*FS	*FS, N	*1.0	*100	*FS	*100	*FS	*FS	*FS, N	*1.0, 0	0	N	0	N	N	N, N	0, 0
catalogue consulting	0	N	0	N	N	N, N	0, 0	0	N	0	N	N	N, N	0, 0	*100	*FS	*100	*FS	*FS	*FS, N	*1.0, 0
pick available item	*100	*FS	*100	*FS	*FS	*FS, N	*1.0, 0	*100	*FS	*100	*FS	*FS	*FS, N	*1.0, 0	*100	*FS	*100	*FS	*FS	*FS, N	*1.0, 0
pre-order non-available item	0	N	0	N	N	N, N	0, 0	0	N	0	N	N	N, N	0, 0	0	N	0	N	N	N, N	0, 0
classic communication handled	*100	*FS	*100	*FS	*FS	*FS, N	*1.0, 0	0	N	0	N	N	N, N	0, 0	0	N	0	N	N	N, N	0, 0
standard form order	0	N	0	N	N	N, N	0, 0	*100	*FS	*100	*FS	*FS	*FS, N	*1.0, 0	*100	*FS	*100	*FS	*FS	*FS, N	*1.0, 0
secure form order	0	N	0	N	N	N, N	0, 0	0	N	0	N	N	N, N	0, 0	0	N	0	N	N	N, N	0, 0
monitoring system	*100	*FS	*100	*FS	*FS	*FS, N	*1.0, 0	*100	*FS	*100	*FS	*FS	*FS, N	*1.0, 0	*100	*FS	*100	*FS	*FS	*FS, N	*1.0, 0
produce statistics	*100	*FS	*100	*FS	*FS	*FS, N	*1.0, 0	*100	*FS	*100	*FS	*FS	*FS, N	*1.0, 0	*100	*FS	*100	*FS	*FS	*FS, N	*1.0, 0
system evolution	*100	*FS	*100	*FS	*FS	*FS, N	*1.0, 0	*100	*FS	*100	*FS	*FS	*FS, N	*1.0, 0	*100	*FS	*100	*FS	*FS	*FS, N	*1.0, 0
add item	*100	*FS	*100	*FS	*FS	*FS, N	*1.0, 0	*100	*FS	*100	*FS	*FS	*FS, N	*1.0, 0	*100	*FS	*100	*FS	*FS	*FS, N	*1.0, 0
check out	*100	*FS	*100	*FS	*FS	*FS, N	*1.0, 0	*100	*FS	*100	*FS	*FS	*FS, N	*1.0, 0	*100	*FS	*100	*FS	*FS	*FS, N	*1.0, 0
update catalogue	*100	*FS	*100	*FS	*FS	*FS, N	*1.0, 0	*100	*FS	*100	*FS	*FS	*FS, N	*1.0, 0	*100	*FS	*100	*FS	*FS	*FS, N	*1.0, 0
check authentication	*100	*FS	*100	*FS	*FS	*FS, N	*1.0, 0	*100	*FS	*100	*FS	*FS	*FS, N	*1.0, 0	*100	*FS	*100	*FS	*FS	*FS, N	*1.0, 0
check information flow	*100	*FS	*100	*FS	*FS	*FS, N	*1.0, 0	*100	*FS	*100	*FS	*FS	*FS, N	*1.0, 0	*100	*FS	*100	*FS	*FS	*FS, N	*1.0, 0
check access control	*100	*FS	*100	*FS	*FS	*FS, N	*1.0, 0	*100	*FS	*100	*FS	*FS	*FS, N	*1.0, 0	*100	*FS	*100	*FS	*FS	*FS, N	*1.0, 0
update GUI	*100	*FS	*100	*FS	*FS	*FS, N	*1.0, 0	*100	*FS	*100	*FS	*FS	*FS, N	*1.0, 0	*100	*FS	*100	*FS	*FS	*FS, N	*1.0, 0
shopping cart	100	FS	100	FS	FS	FS, N	1.0, 0	100	FS	100	FS	FS	FS, N	1.0, 0	100	FS	100	FS	FS	FS, N	1.0, 0
select items	100	FS	100	FS	FS	FS, N	1.0, 0	100	FS	100	FS	FS	FS, N	1.0, 0	100	FS	100	FS	FS	FS, N	1.0, 0
get identification details	100	FS	100	FS	FS	FS, N	1.0, 0	100	FS	100	FS	FS	FS, N	1.0, 0	100	FS	100	FS	FS	FS, N	1.0, 0
internet managed	0	N	0	N	N	N, N	0, 0	100	FS	100	FS	FS	FS, N	1.0, 0	100	FS	100	FS	FS	FS, N	1.0, 0
perform privacy control	**90**	**PS**	**75**	**(FS)**	**(FS)**	**PS, N**	**0.5, 0**	**90**	**PS**	**75**	**(FS)**	**(FS)**	**PS, N**	**0.5, 0**	**90**	**PS**	**75**	**(FS)**	**(FS)**	**PS, N**	**0.5, 0**
manage internet orders	100	FS	100	FS	FS	FS, N	1.0, 0	100	FS	100	FS	FS	FS, N	1.0, 0	100	FS	100	FS	FS	FS, N	1.0, 0
adaption	100	FS	100	FS	FS	FS, N	1.0, 0	100	FS	100	FS	FS	FS, N	1.0, 0	100	FS	100	FS	FS	FS, N	1.0, 0
manage internet searching	100	FS	100	FS	FS	FS, N	1.0, 0	100	FS	100	FS	FS	FS, N	1.0, 0	100	FS	100	FS	FS	FS, N	1.0, 0
manage internet shop	100	FS	100	FS	FS	FS, N	1.0, 0	100	FS	100	FS	FS	FS, N	1.0, 0	100	FS	100	FS	FS	FS, N	1.0, 0
privacy	**75**	**PS**	**75**	**FS**	**FS**	**PS, N**	**0.5, 0**	**40**	**N**	**50**	**(PS)**	**(C)**	**PS, PD**	**0.5, 0.5**	**40**	**N**	**50**	**(PS)**	**(C)**	**PS, N**	**0.5, 0**
availability	25	PS	25	PS	**(FS)**	PS, N	0.5, 0	50	PS	25	PS	**(FS)**	PS, N	0.5, 0	50	PS	25	PS	**(FS)**	PS, N	0.5, 0
integrity	**0**	**N**	**0**	**(C)**	**(C)**	**PS,PD**	**0.5, 0.5**	**0**	**N**	**0**	**(C)**	**(C)**	**PS, PD**	**0.5, 0.5**	50	PS	25	PS	**(C)**	PS, N	0.5, 0
usability	**25**	**PS**	**25**	**(C)**	**(C)**	**PS, N**	**0.5, 0**	**47**	**PS**	**24**	**(C)**	**(C)**	**PS, N**	**0.5, 0**	90	PS	37	(PS)	**(C)**	PS, N	0.5, 0
adaptability	100	FS	100	FS	FS	FS, N	1.0, 0	100	FS	100	FS	FS	FS, N	1.0, 0	100	FS	100	FS	FS	FS, N	1.0, 0
easy to use	-25	PD	-25	PD	**(FD)**	N, PD	0, 0.5	-50	PD	-25	PD	**(FD)**	N, PD	0, 0.5	50	PS	25	PS	**(FS)**	PS, N	0.5, 0
security	24	PS	25	(PS)	**(C)**	PS, N	0.25, 0	45	PS	19	(PS)	**(C)**	PS, N	0.25, 0	70	PS	25	(PS)	**(C)**	PS, N	0.25, 0

Table 46: Results of the Evaluation Techniques Applied for Three Alternatives over the Wireless Services Model in Figure 10

		Alternative 1							Alternative 2							Alternative 3						
Actor	Element	*GRL-Quant*	*GRL-Qual*	*GRL-Hybrid*	*i**	*NFR*	*Tropos-Qual*	*Tropos-Quant*	*GRL-Quant*	*GRL-Qual*	*GRL-Hybrid*	*i**	*NFR*	*Tropos-Qual*	*Tropos-Quant*	*GRL-Quant*	*GRL-Qual*	*GRL-Hybrid*	*i**	*NFR*	*Tropos-Qual*	*Tropos-Quant*
Service Provider	Low Cost	37	PS	10	PS	**(FS)**	PS, N	0.25, 0	37	PS	10	PS	**(FS)**	PS, N	0.125, 0	**-12**	**N**	**-3**	**C**	**C**	**PS, N**	**0.125, 0**
	High Performance	**0**	**N**	**0**	**C**	**(C)**	**PS, N**	**0.25, 0**	**25**	**N**	**19**	**C**	**C**	**PS, N**	**0.25, 0**	**25**	**N**	**19**	**C**	**C**	**PS, N**	**0.25,0**
	Minimum Changes to Infrastructure	75	PS	38	(PS)	**(FS)**	PS, N	0.5, 0	**75**	**PS**	**38**	**(PS)**	**(FS)**	**PS, PD**	**0.25, 0.5**	**-25**	**N**	**-12**	**(C)**	**(C)**	**PS, PD**	**0.25, 0.5**
	Maximum Hardware Utilization	*50	*PS	*50	*PS	*PS	*PS, N	*0.5, 0	*50	*PS	*50	*PS	*PS	*PS, N	*0.5, 0	*50	*PS	*50	*PS	*PS	*PS, N	*0.5, 0
	High Throughput	**0**	**N**	**0**	**(C)**	**(C)**	**PS, N**	**0.5, 0**	**25**	**N**	**19**	**(C)**	**(C)**	**PS, N**	**0.5, 0**	**25**	**N**	**19**	**(C)**	**(C)**	**PS, N**	**0.5, 0**
	Minimum Message Exchange	100	FS	100	FS	FS	FS, N	1.0, 0	-50	PD	-25	PD	**(FD)**	N, PD	0, 0.5	-50	PD	-25	PD	**(FD)**	N, PD	0, 0.5
	Minimum Switch Load	-100	FD	-100	FD	FD	N, FD	0, 1.0	100	FS	100	FS	FS	FS, N	1.0, 0	100	FS	100	FS	FS	PS, N	1.0, 0
System	Determine Data Location	100	FS	100	FS	FS	FS, N	1.0, 0	**0**	**N**	**0**	**N**	**N**	**FS, N**	**1.0, 0**	100	FS	100	FS	FS	FS, N	1.0, 0
	Data in Service Control Point	*100	*FS	*100	*FS	*FS	*FS, N	*1.0, 0	0	N	0	N	N	N, N	0, 0	0	N	0	N	N	N, N	0, 0
	Data in New Service Node	0	N	0	N	N	N, N	0, 0	**-100**	**FD**	**-100**	**FD**	**FD**	**FS, N**	**1.0, 0**	100	FS	100	FS	FS	FS, N	1.0, 0
	Service in Central Switch	*100	*FS	*100	*FS	*FS	*FS, N	*1.0, 0	0	N	0	N	N	N, N	0, 0	0	N	0	N	N	N, N	0, 0
	Service in Service Control Point	0	N	0	N	N	N, N	0, 0	*100	*FS	*100	*FS	*FS	*FS, N	*1.0, 0	*100	*FS	*100	*FS	*FS	*FS, N	*1.0, 0
	Determine Service Location	100	FS	100	FS	FS	FS, N	1.0, 0	100	FS	100	FS	FS	FS, N	1.0, 0	100	FS	100	FS	FS	FS, N	1.0, 0
	Install Service Node	0	N	0	N	N	N, N	0, 0	*100	*FS	*100	*FS	*FS	*FS, N	*1.0, 0	*100	*FS	*100	*FS	*FS	*FS, N	*1.0, 0
Vendor	Service Nodes Ready for Sale	0	N	0	N	N	N, N	0, 0	*-100	*FD	*-100	*FD	*FD	*N, FD	*0, -1.0	*100	*FS	*100	*FS	*FS	*FS, N	*1.0, 0
Depend-encies	Service Node	0	N	0	N	N	N, N	0, 0	**-100**	**FD**	**-100**	**FD**	**FD**	**N, N**	**0, 0**	*100	*FS	*100	*FS	*FS	*FS, N	*1.0, 0

Table 47: Results of the Evaluation Techniques Applied for Three Alternatives over the Counseling Services Model in Figure 11

		Alternative 1							Alternative 2							Alternative 3						
Actor	Element	*GRL-Quant*	*GRL-Qual*	*GRL-Hybrid*	*i**	*NFR*	*Tropos-Qual*	*Tropos-Quant*	*GRL-Quant*	*GRL-Qual*	*GRL-Hybrid*	*i**	*NFR*	*Tropos-Qual*	*Tropos-Quant*	*GRL-Quant*	*GRL-Qual*	*GRL-Hybrid*	*i**	*NFR*	*Tropos-Qual*	*Tropos-Quant*
Counsellors	Use Text messaging	*100	*FS	*100	FS	FS	FS, N	1.0, 0	*-100	*FD	*-100	FD	FD	N, FD	0, 1.0	*100	*FS	*100	FS	FS	FS, N	1.0, 0
	Use Cyber Café/Portal/Chat Room	*-100	*FD	*-100	FD	FD	N, FD	0, 1.0	*100	*FS	*100	FS	FS	FS, N	1.0, 0	*100	*FS	*100	FS	FS	FS, N	1.0, 0
	Provide Online Counseling Services	100	FS	100	FS	FS	FS, N	1.0, 0	100	FS	100	FS	FS	FS, N	1.0, 0	100	FS	100	FS	FS	FS, N	1.0, 0
	Help as Many Kids as Possible	50	PS	25	PS	(FS)	PS, N	0.5, 0	50	PS	25	PS	(FS)	PS, N	0.5, 0	50	PS	25	PS	(FS)	PS, N	0.5, 1
	Listen for cues	-50	PD	-75	(FD)	(C)	N, FD	0, 1.0	50	PS	75	(PD)	(C)	N, PD	0, 0.5	-100	FD	-100	FD	(FD)	N, FD	0, 1.0
	High quality counseling	-50	PD	-25	(PD)	(C)	N, PD	0, 0.25	0	PD	13	(PD)	(C)	N, PD	0, 0.25	-75	PD	-31	(PD)	(FD)	N, PD	0, 0.25
	Avoid burnout	-25	PD	-6	PD	(FD)	N, PD	0, 0.25	-25	PD	-6	PD	(FD)	N, PD	0, 0.25	-25	PD	-6	PD	(FD)	N, PD	0, 0.25
	Happiness [counselors]	-12	PD	-1	(C)	(C)	PS, N	0.25, 0	13	N	8	(C)	(C)	PS, N	0.25, 0	-24	PD	-3	(C)	(C)	PS, N	0.25, 0
Kids and youth	Use text messaging	0	N	0	FS	FS	FS, N	1.0, 0	-100	D	-100	FD	FD	N, N	0, 0	0	N	0	FS	FS	FS, N	1.0, 0
	Kids use cyber café/portal/chat room	-100	FD	-100	FD	FD	N, N	0, 0	0	N	0	FS	FS	FS, N	1.0, 0	0	N	0	FS	FS	FS, N	1.0, 0
	Help be aquired	0	N	0	FS	FS	FS, N	1.0, 0	0	N	0	FS	FS	FS, N	1.0, 0	0	N	0	FS	FS	FS, N	1.0, 0
	Comfortableness with service	-25	N	-19	(PS)	(C)	PS, N	0.5, 0	-62	PD	-31	(C)	(C)	PS, N	0.5, 0	0	N	0	(PS)	(C)	PS, N	0.5, 0
	Anonymity [service]	50	PS	25	(PS)	(FS)	PS, N	0.5, 0	-25	PD	-25	(PD)	(FD)	N, PD	0, 0.5	0	N	0	(C)	(C)	PS, PD	0.5, 0.5
	Immediacy [service]	-100	FD	-100	FD	(FD)	N, PD	0, 0.5	25	PS	25	FS	(FS)	FS, N	1.0, 0	0	N	0	(PS)	(C)	FS, PD	1.0, 0.5
	Get effective help	-37	N	-24	(C)	(C)	PS, N	0.25, 0	-31	PD	-8	(PD)	(C)	PS, N	0.5, 0	0	N	0	(PS)	(C)	PS, N	0.5, 0
Organization	Use Cyber Café/Portal/Chat Room	-100	FD	-100	FD	FD	N, N	0, 0	0	N	0	FS	FS	FS, N	1.0, 0	0	N	0	FS	FS	FS, N	1.0, 0
	Use Text messaging	0	N	0	FS	FS	FS, N	1.0, 0	-100	D	-100	FD	FD	N, N	0, 0	0	N	0	FS	FS	FS, N	1.0, 0
	Provide Online Counseling Services	0	N	0	FS	FS	FS, N	1.0, 0	0	N	0	FS	FS	FS, N	1.0, 0	0	N	0	FS	FS	FS, N	1.0, 0
	Immediacy [service]	-50	PD	-25	(PD)	(FD)	N, PD	0, 0.5	50	PS	25	(PS)	(FS)	PS, N	0.5, 0	0	N	0	(C)	(C)	PS, PD	0.5, 0.5
	Anonymity [service]	50	PS	25	(PS)	(FS)	PS, N	0.5, 0	-50	PD	-25	(PD)	(FD)	N, PD	0, 0.5	0	N	0	(C)	(C)	PS, PD	0.5, 0.5
	Help as Many Kids as Possible	-6	PD	-1	(C)	(C)	PS, N	0.5, 0	-6	PD	0	(C)	(C)	PS, N	0.5, 0	-18	PD	-2	(C)	(C)	PS, N	0.5, 0
	High quality counseling	-50	PD	-25	(FD)	(C)	N, FD	0, 1.0	0	N	0	(PD)	(C)	PS, PD	0.25, 0.5	-75	PD	-31	(FD)	(C)	PS, FD	0.25, 1.0
	Avoid scandal	25	PS	6	PS	(FS)	PS, N	0.25, 0	-25	PD	-6	PD	(FD)	N, N	0, 0	0	N	0	(C)	(C)	PS, N	0.25, 0
	Increase funds	-13	N	-4	(C)	(C)	PS, N	0.125, 0	-12	PD	-1	(FD)	(C)	PS, N	0.125, 0	-37	PD	-8	(PD)	(C)	PS, N	0.125, 0
	Help kids	-28	PD	-6	(PD)	(C)	PS, N	0.25, 0	-3	PD	0	(PD)	(C)	PS, N	0.25, 0	-46	PD	-8	(PD)	(C)	PS, N	0.25, 0
Dependencies	Help as Many Kids as Possible	0	N	0	PS	FS	PS, N	0.5, 0	0	N	0	PS	FS	PS, N	0.5, 0	0	N	0	PS	FS	PS, N	0.5, 0
	High quality counseling	-50	PD	-25	PD	C	N, N	0, 0	0	N	0	PD	(C)	N, N	0, 0	-75	PD	-31	PD	FD	N, N	0, 0
	Provide counselling via text message	0	N	0	FS	FS	FS, N	1.0, 0	-100	FD	-100	FD	FD	N, N	0, 0	0	N	0	FS	FS	FS, N	1.0, 0
	Provide counseling via cyber café/portal	-100	FD	-100	FD	FD	N, N	0, 0	0	N	0	FS	FS	FS, N	1.0, 0	0	N	0	FS	FS	FS, N	1.0, 0
	Text messaging service	0	N	0	FS	FS	FS, N	1.0, 0	-100	FD	-100	FD	FD	N, N	0, 0	0	N	0	FS	FS	FS, N	1.0, 0
	Cyber café/portal/chat room	-100	FD	-100	FD	FD	N, N	0, 0	0	N	0	FS	FS	FS, N	1.0, 0	0	N	0	FS	FS	FS, N	1.0, 0

4.7 Analysis

Examining the analysis results over all the models, we can see that each alternative over each model produces differences in results. The differences are often significant. The observed differences include differing values for the top-level goals. For example in Alt 1, Help Kids in Organization is partially satisfied, partially denied and conflicted in the results of three different procedures. We also see that the two types of numeric procedures (GRL, Tropos) differ in how they resolve and combine numbers, sometimes resulting in drastically different results (see for example, Help as Many Kids as Possible in Organization). We analyze the differences first on a more detailed level, with the purpose of understanding why these differences occur. Then we analyze the differing choices between alternatives that these differences may cause.

Result Differences. We count the number of model elements which have differing results for each alternative, with the results summarized in the first column (All) for each alternative in Table 48. For example, in the Alternative 1 columns of Table 46 (Wireless Service Model), seven of the elements have results which differ, where one or more of the results for the seven procedures were bolded in a total of seven rows. This is reflected in the Wireless Service Model row in Table 48.

Table 48: Count of the Number of Differences in the Results over all Model Elements per Alternative

	Alt1				Alt2				Alt3				Totals
	All	Dep	Sym	SG	All	Dep	Sym	SG	All	Dep	Sym	SG	All
Media Shop	4	0	1	3	8	0	4	4	5	0	1.5	3.5	17
Wireless Service	7	0	0.5	6.5	7	0	0.5	6.5	7	0	0.5	6.5	21
Counseling Service	28	7	10.5	10.5	27	7	8	12	27	11	8.5	8	82
Totals	39	7	12	20	42	7	12.5	22.5	39	11	11	18	120

We can attribute these differences to a combination of analysis procedural choices and structural characteristics of the three models. We can identify three procedural choices which result in the majority of the result differences: (1) the different treatment of dependency values (Dep) -- GRL treats them as constraints while the other procedures treat them as requirements; (2) the asymmetry of links (Sym) -- the Tropos procedures are not propagating negative values; and (3) different methods for the resolution of values for softgoals (SG) -- the NFR procedure insists that final values must not be partial, the NFR and i* procedures allow value promotion, and the GRL does not include the concept of a conflict. We count the number of differences which are caused by each of these three choices (Dep, Sym, and SG, respectively), displaying these counts in Table 48. In counting the choices, sometimes results for one alternative are caused by more than one procedure choice. In this case we divide the count by the number of causes, so that the sum of the individual counts adds up to the total number of elements with differences. For example, looking at the seven differences in the first alternative of the media shop model, six differences are caused by differences in the way softgoals are resolved, while one difference, for the softgoal adaptability, is caused in part by differences in softgoal resolution (PD vs. FD for i* vs. NFR) and in part by the asymmetry of links, with negative values from easy to use not propagated to this softgoal. We can observe from the sum in the last column of Table 48 that different methods for softgoal resolution (SG) accounts for a slightly larger percentage of the result differences when compared to different methods for dependency propagation (Dep) or link symmetry (Sym).

The number of differences in the results for the first two models is similar, ranging between 4 and 7. These models have a similar structure in that the links are mostly AND/OR links, the models have little or no dependency links, and there are few softgoals when compared to the Counseling model. Statistics concerning the elements and links in each model can be seen in Table 49 and Table 50. Presentation of the Media Shop model in (Giorgini et al., 2005) distinguishes between soft and hard goals, although presentation in the tool, as reflected in Figure 9, does not. Generally, in this model, all goals with incoming contribution links are softgoals. We can examine the ratio of Softgoals to Goals for each of the three models: 0.27, 2.33 and 6.0, for the Media, Wireless, and Counseling Models, respectively. In other words, there are roughly 4 goals for every softgoal for the Media model, 2 softgoals for every goal in the wireless service, and 6 softgoals for every goal in the counseling model. Because we have identified softgoal resolution (SG) as one of the most significant causes of result differences, models with more softgoals will have a greater divergence. Similarly, we can look at the ratio of contribution to AND/OR links: 0.95, 1.43, and 5.4, respectively. The Media Shop model has a roughly equal number of contribution and AND/OR links, while the Wireless Model has about 1.5 times more contribution links, and the counseling model has more than 5 contribution links for every AND/OR link in the model. The Counseling Model also has many more dependencies (12 vs. 2 or 0) when compared to the other models. Similar to the presence of softgoals, our results show that models with many contributions (Sym) or dependency (Dep) links are also more likely to produce different results between procedures.

Table 49: Element Statistics for each of the three Example Models

	Goals	Softgoals	Tasks	Resources	Total
Media Shop	26	7	0	0	33
Wireless Service	3	7	5	1	16
Counseling Service	3	18	8	2	31

Table 50: Link Statistics for each of the three Example Models

	AND	OR	Depend.	Make	Some+	Help	Hurt	Some-	Break	Total
Media Shop	12	8	0	3	0	13	3	0	0	39
Wireless Service	3	4	2	2	4	1	1	1	1	19
Counseling Service	0	6	12	1	0	22	8	0	2	51

Alternative Selection. Although the examination of differences at a detailed level is interesting, the alternative (solution) that the results would lead evaluators to select is more important. Several of the selected analysis procedures emphasize use of analysis procedures to evaluate the impact of decisions over alternatives (Chung et al., 2000), select an alternative (Horkoff, 2006), or evaluate an alternative (Amyot et al., 2010; Giorgini et al., 2005). The purpose of these activities is to make a selection over one or more high-level design alternative in the model. We use the results of our comparisons of the selected procedures over the example models in order to select one alternative set of design options in each model. These alternatives have been listed in Section 4.3.

We rank the alternatives for each model (1st, 2nd or 3rd), using each of the seven procedures, presented in Table 51. In this table, the rows represent the selection result for each procedure, while the column represents the model and alternative. For example, the third column (Wireless Service, Alt 2), fifth row (GRL-Mixed) shows that after applying the GRL-Mixed procedure to Alternatives 2 in the Wireless Service model, we would select this alternative as the first choice out of the three alternatives for this model. Similarly, Alt 3 would be the second choice, and Alt 1 the third choice. As there is no systematic way in any method to select an alternative given the goal evaluation results, we pick the alternative with the most strongly satisfied criteria (softgoals, high-level hard goals). For the quantitative procedures, we decide to select an alternative by adding the results over criteria elements. The sums for the criteria goals are listed in parenthesis after the rank in Table 51. In our example, the sum is 161 for Alt 2, 98 for Alt 3 and 48 for Alt 1. Note that this approach is not recommended by any of the authors in (Giorgini et al., 2005) or (Amyot et al., 2010); however the presence of numbers makes this approach feasible and tempting.

The table contains several entries with "or" or with "?". In these cases it is difficult to make a ranking decision over qualitative values for each alternative which generally show trade-offs over several goals. For example, Alt 1 and 2 results for the NFR procedure in the Wireless Service model differ by two goals; either Minimum Message Exchange is FS while Minimum Switch Load is FD, or the inverse. It is difficult to make a decision in this case, without further information, such as the relative priorities of these goals. We have indicated this difficulty in Table 51 by indicating that Alt 1 and 2 for this model (Wireless Service) and procedure (NFR) can be each ranked either first or second. In the Counseling Service model, none of the alternatives satisfied the important goals of each actor sufficiently. As a result, we feel it is not possible to make a selection over the available alternatives, looking at the qualitative results over this model an analyst would likely suggest that none of the alternatives be selected. We indicate this in Table 51 by placing "?" instead of a ranking amongst alternatives. In these cases the model could be refined to include further criteria or further alternatives could be suggested.

Table 51: Ranking of Alternatives for each Model Based on Analysis Results

	Wirless Service			Media Shop			Counseling Service		
	Alt 1	Alt 2	Alt 3	Alt 1	Alt 2	Alt 3	Alt 1	Alt 2	Alt 3
GRL-Quant	2nd (112)	1st (212)	3rd (63)	3rd (224)	2nd (232)	1st (450)	2nd (-296)	1st (-76)	3rd (-375)
GRL-Qual	1st or 2nd	1st or 2nd	3rd	2nd	3rd	1st	?	?	?
GRL-Mixed	3rd (48)	1st (161)	2nd (98)	2nd (225)	3rd (193)	1st (287)	3rd (-249)	1st (50)	2nd (-183)
*i**	1st	2nd	3rd	2nd	3rd	1st	2nd	3rd	1st
NFR	1st or 2nd	1st or 2nd	3rd	2nd	3rd	1st	?	?	?
Tropos-Qual	1st	2nd or 3rd	2nd or 3rd	2nd	3rd	1st	?	?	?
Tropos-Quant	1st (3.0, 1.0)	2nd or 3rd (2.625, 1.0)	2nd or 3rd (2.625, 1.0)	2nd (3.25, 1.0)	3rd (3.25, 1.5)	1st (3.75, 0)	3rd (1.25, 2.0)	1st (1.75, 1.5)	2nd (3.0, 3.0)

In the Media Shop model, the selection of alternatives is generally consistent, with the exception of the first GRL-Quant results. In the Wireless Service model, we see more ranking differences, especially with the GRL-Mixed procedure. These differences demonstrate that differing decisions over procedure conventions have the potential to produce differing alternative selections – even in models with few softgoals, contributions, or dependency links. The selections for the Counseling Service model also differ between procedures, not surprisingly considering the differences in analysis results. In this model it is often difficult to make decisions over alternatives, especially when analyzing qualitative results.

4.8 Discussion

In this section, we discuss the impact of our comparison results, including the perceived reliability of analysis results (R14), the benefits of forward satisfaction analysis procedures, and how results can shape the use of such procedures in practice. We consider additional factors which may have further affected comparison results, such as goal priorities, model size, and inconsistent human judgment. Finally, we consider threats to the validity of our comparison study and the generalizability of the results to other goal model analysis procedures.

The results of our comparison in Section 4.6 show that the selected procedures produce significantly varying results when analyzing the selected alternatives over the sample models. We see that differing assumptions concerning goal concepts and propagation can have a noticeable effect on analysis results. We believe that the presence of the differing results is significant. Depending on the analysis procedure selected, potential users could make widely different conclusions about the effects of system alternatives on domain goals. These conclusions could lead to differing selections over functional or design alternatives represented in the models.

For example, in Table 51, if results from the analysis procedures were followed without question, the GRL-Mixed analysis procedure would lead users to select Alternative 2, where the service is in the service control point and the data is in a new service node. Results from the i* procedure would lead users to select Alternative 1, where the service is in the central switch and where the new service node is not installed. These are very different decisions using the same criteria in the same model.

Our results have shown that the structure of the underlying model makes a significant difference in the consistency between analysis results. Specifically, the presence of many softgoals, dependencies, and contribution links decreases confidence in analysis results. As softgoals and contribution links are intended to represent "fuzzy", flexible concepts and effects which are difficult to quantify or formalize, and dependencies represent often represent high-level, social interactions, it is reasonable that different interpretations of these constructs causes differences between analysis results. However, these social and non-functional constructs are particularly useful in early RE analysis, as discussed in Section 1.1.

We can observe that making decisions over models with many "criteria" goals can be especially difficult, as shown with the high number of '?' in Table 51 for alternative selections in the Counseling Service model. Results in Table 51 show that quantitative values give a more fine-grained evaluation of the results when compared to qualitative procedures, and that this granularity helps to distinguish between alternatives. These results could lead to a recommendation of quantitative procedures such as Tropos-quant or GRL-quant for decision making between alternatives. However, differences in quantitative interpretations of goal model constructs can produce differing results, as seen in Table 51. There is a danger in placing too much precision in such numbers, which can actually be thought of as finer-grained qualitative estimates. Operations such as addition, although tempting, lead to a false sense of precision. We observe a fundamental trade-off between qualitative and quantitative procedures. Qualitative procedures lack precision and can make selecting an alternative over multiple criteria goals difficult. Quantitative procedures are more precise, which can help to better differentiate between alternatives; however, numbers are affected by procedure design choices, such as conversions from qualitative to quantitative and choices over propagation rules. Furthermore, it is tempting but ill-advised to perform mathematical operations, such as addition, over these results. The precision offered by quantitative procedures needs to be taken with a grain of salt, treating numbers as estimates, with uncertainty increasing with subsequent propagation. The difference between 0.5 and 0.25 may be helpful, but the difference between 0.07 and 0.10 is likely not significant.

The lesson from these observations should not be that goal models should not be analyzed systematically, but that when used in decision making, the analysis process and results should be considered as a heuristic, and always be interpreted in the context of the domain. In fact, the process of modeling and evaluating is likely as useful as the results, as the process forces the evaluator to examine the model and their domain knowledge and assumptions. Although analysis results for "softer", more social models (softgoals, contributions, dependencies) show a greater variation than "harder" models (AND/OR, hard goals), users should not be discouraged from performing analysis over softer models. In fact, the fuzzy nature of these models calls for systematic analysis in a process of elicitation and specification. However, the role of this analysis should be clear. The variances in the comparison results from Section 4.7 emphasize that the benefits of forward satisfaction techniques may lie less in their ability to facilitate decision making, and more in their ability to provide other benefits, such as improved domain understanding, communication, scoping, and elicitation. For softer, more social models, interactive analysis can encourage a learning process, forcing users to question their assumptions about the model and domain, evolving the model from a draft to a sufficient level of completeness and accuracy. When this process is performed in a group setting, it can facilitate interesting discussions which promote convergent understanding. We test these assumptions with several empirical studies in Chapter 12.

Evolution over softer models can lead to a more clearly specified, more stable, harder model, over which analysis results can be more reliable. In fact, if there is enough available time, goal model users could apply both interactive qualitative, and then automatic quantitative analysis as part of the same process. We can describe such an approach using the concept of early and late RE from the i* and Tropos approaches (Bresciani et al., 2004; E. Yu, 1997). These stages could lead into a design of system architecture, as in (Bresciani et al., 2004). See Figure 13 for an example high-level development process using both styles of analysis.

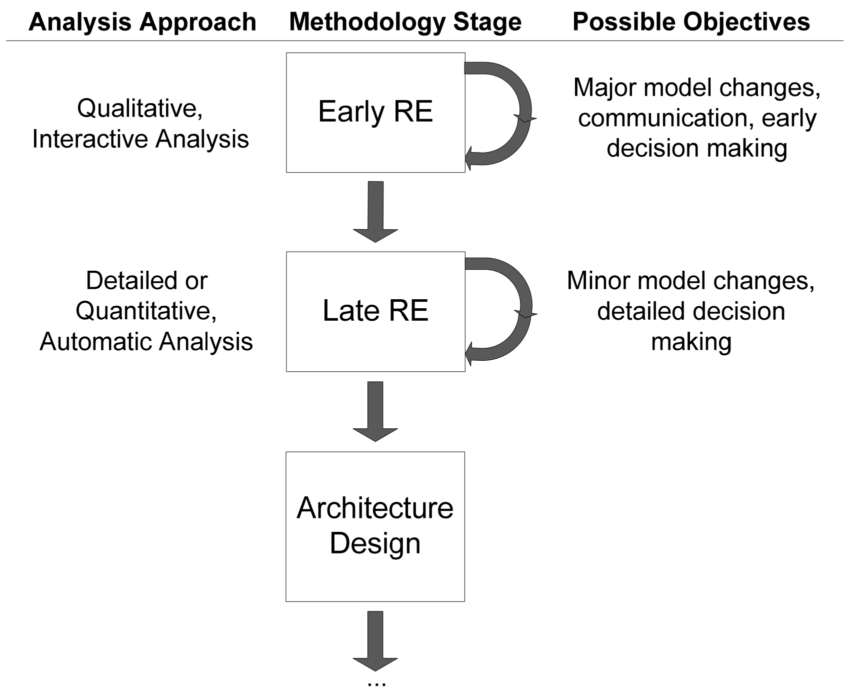


Figure 13: Example development process using both qualitative and quantitative goal model analysis

Both the i* and NFR analysis procedures specifically aim for early RE, supporting high-level understanding and decision making in the absence of specific metrics. The GRL and Tropos approaches leave application open to either early or late RE stages. However, it is difficult to acquire formal or quantified domain information in the early stages of understanding, especially if aiming for completeness. Therefore, these procedures are likely better suited to later stage models, where more detailed information is available. After qualitative, interactive evaluation such as the i* or NFR procedures have been applied, during later stages of requirements analysis, more precise, quantitative forms of evaluation such as the Tropos-quant or GRL-quant can be applied to relatively stable models, producing results which are likely to be more detailed and accurate, and which may be better suited to decision making. Later stages of analysis could incorporate partial or complete system metrics into the analysis process, such as done in (Barone et al., 2011; van Lamsweerde, 2009; Pourshahid et al., 2011), potentially increasing the accuracy of analysis results. We give further consideration to modeling and analysis methodologies in Chapter 8.

4.8.1 Threats to Validity

Although the analysis of results over the three sample models has produced useful observations, there are some threats to the validity of our study and comparison. First the analysis was only performed over three models. Although an effort was made to pick models which had differing structures covering different domains, further analysis over more sample models is needed to increase confidence in our results.

The models used in our comparison were of small to medium size to facilitate comprehension of results. If larger models were used, the variance in results may be even larger, as larger models mean longer propagation paths away from the common starting point of initial alternative labels. The heuristic nature of forward satisfaction analysis procedures should be especially emphasized for larger models.

Manual classification of the underlying reasons for results differences could be subject to error and interpretation. In fact, differences are often propagated; an initial difference in results for one intention is propagated to produce differences for further intentions. In this case, the cause for the difference was counted once for each intention where it appeared, whether that intention originated the results difference or not. We felt that this convention provided the greatest measure of the impact of individual procedure choices causing results differences.

The ranking of alternatives using the qualitative results of each procedure was conducted manually, creating an area of potential contention. The rankings were performed by trying to select the most obviously favorable alternative according to results over multiple intentions, giving an implicit priority to intentions towards the "top" of the model. Although this ranking may change if performed by other individuals using different criteria, especially those with more in depth knowledge of the model domain, obviously unclear choices were marked with a "?", making the decision less arbitrary. In general, there is a need for methods which allow comparing the results of multiple satisfaction analysis runs, allowing users to select an appropriate alternative.

When running analysis procedures over models we chose to use the implementation available in associated software tools. The available tools are produced and maintained by research institutions, and are not commercialized. It is possible that the implementations may produce erroneous or unexpected results. Whenever possible our results were checked against results available in the originating papers, with differences analyzed and explained. An example of unexpected results was the presence of only asymmetric propagation in the GR-Tool, when the associated paper clearly meant for the symmetry of contribution links to be manually specified and default to symmetric behavior. We could have chosen to manually propagate evidence as specified in the paper and not as implemented in the tool; however, doing so may have introduced errors in either the manual propagation or our interpretation of the procedure, especially for the quantitative propagation. We chose to follow the implemented procedure in all cases.

Comparison of results required an introduction of conversion and conventions as described in Section 4.5. It is possible that such conventions may affect the comparison and classification of our results. However, we believe that these affects are minimal. For example, using a convention of PS = 10 to 90 instead of PS = 5 to 94 would have a nominal affect on our difference counts.

Certain factors, if included in the design of our comparison, may have had a further impact on results. For example, in our comparison, we have not made use of explicit measures for goal importance available in some of the procedures, e.g., (Amyot et al., 2010); however, the implicit importance of goals captured in the positioning of the model can be helpful. For example, in Figure 11 there is obviously a “top” goal for each actor, and these goals could be given more weight when making decisions.

When applying human judgment in applicable procedures, we chose resulting labels consistent between procedures, and consistent with the structure of the model. However, results of user studies using the i* procedure found that users often made decisions inconsistent with the structure of the underlying model. For example a set of incoming labels {PS, PS, PS} may be combined together as a Conflict (C). These results are described further in Chapter 12. If users make inconsistent judgments, the variance between the results of interactive and automatic procedures would vary even more significantly, as automatic procedures make automated decisions which reflect their assumptions over the underlying model concepts. Chapter 10 describes mechanisms to make users aware of the inconsistencies between their judgments and the structure of the model. The occurrence of inconsistent judgments further emphasizes the use of forward satisfaction procedures as a means of model improvement and domain understanding, as opposed to a decision making tool.

Although we have attempted to be neutral in our analysis, we are comparing existing procedures to the procedure described in Chapter 5. Our intention is not to declare one or more techniques as superior, but to understand the role of qualitative, forward goal model analysis as a decision making tool, to test the reliability of analysis results, and to help goal modelers to select appropriate analysis approaches.

4.8.2 Generalizability

In this chapter, we have compared qualitative analysis procedures which propagate evidence in a forward direction. Results have shown that these procedures produce results which are inconsistent from each other, especially over highly social models. It is left to consider whether other types of goal-oriented analysis procedures would produce results which are more reliable. Generally, there is a relationship between model completeness, accuracy, precise model interpretation, expressiveness and the accuracy and reliability of results. If models are more complete, and accurate, if the syntax is more precise and expressive, results will be more reliable and accurate. Of course, there are tradeoffs between completeness, accuracy, precision, flexibility, complexity and procedure usability. If the models are more complete, accurate, more precise and more expressive, the modeling syntax will be more complex, the procedures will be less usable, or will at least have a higher learning curve, and the language will be less flexible. These are the tradeoffs that distinguish early RE analysis from later stage analysis: in early RE analysis model flexibility and procedure usability are especially important to involve stakeholders and encourage elicitation and understanding.

Given these considerations, we can generalize our results by assuming that analysis procedures which involve more specific information over more precisely expressed models will produce more reliable results. Conversely, those models that accommodate high-level, social or non-functional aspects represented via softgoals and dependency links may produce results which are less reliable. For example, approaches such as (Franch, 2006; 2009), which calculate metrics over i* models without added precision or expressiveness and without explicitly collecting metrics from the domain, may produce results which are similarly unreliable as the procedures compared in this chapter. Other approaches, such as (Fuxman et al., 2001; Letier & van Lamsweerde, 2004), which use more expressive syntax and require additional information such as probabilities or ordering may produce more reliable results. See Table 38 and Table 39 for more examples of procedures which support softgoals and dependencies and which require more precise information. It is out of the scope of this work to compare and test the reliability of all types of goal analysis procedures.

4.9 Conclusions

Analysis over goal models has been suggested as a means to aid in decision making using objectives captured in the model. In this chapter, we have provided a detailed comparison of forward satisfaction algorithms. We have applied seven sample procedures to three alternatives over three models using three tools. Results show that differing design choices and syntax interpretation amongst procedures can produce differing results over sample models. These results, if taken at face value, can lead users to select different feature or design alternatives captured in the model. The structure of the model to be analyzed plays a significant role in variance between procedure results and in the reliability of results in general. Specifically, the presence of many softgoals, contribution links, or dependencies may make results less reliable. However, these are exactly the syntax constructs which are most useful for early, high-level requirements analysis, the focus of this thesis. Our results deemphasize the use of goal model analysis procedures as a decision making tool, implying that the selected analysis procedures are unable to provide reliable analysis (R14), and limiting their analysis power. The selected procedures used as decision making tools should be heuristic in nature. Our results emphasize use of goal model analysis procedures for other benefits, leading to a focus on their role as a tool to guide domain exploration, including model improvement (completeness and accuracy), improved communication (stakeholder involvement), and capturing rationale. We summarize the results of this chapter by updating analysis of our model summarizing the requirements for analysis of agent-goal models in early RE based on existing work in Figure 13. We have placed an initial value of partially denied for the R14 Reliable Analysis Requirement (previously conflict after making a judgment over incoming labels).

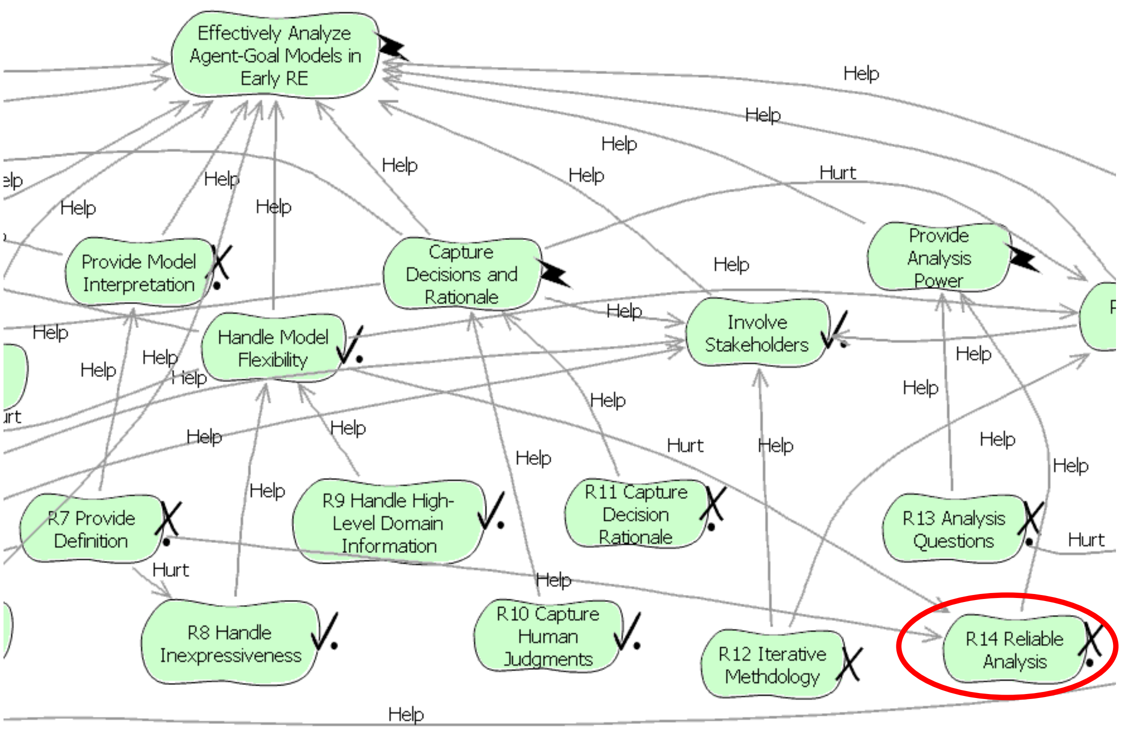


Figure 14: Satisfaction Analysis for the Requirements for analysis of Agent-Goal Models in Early RE based on a Combination of Existing Work Updated using results from Chapter 4

Chapter 5 Reflective Analysis and Formal Definition of the Syntax and Semantics of the i* Agent-Goal Modeling Framework

Section 3.4 describes our selection of the i* Framework as a platform for our iterative, interactive early RE analysis framework. Description of the i* Framework in (E. Yu, 1995; 1997) aimed to define a framework flexible enough to facilitate modeling of early requirements. Although suggestions and potential directions for use of the modeling notation have been provided in (E. Yu, 1995; 1997), the description of the Framework was left open to a certain degree of interpretation and adaptation. Consequently, the framework has been applied to many different areas, including requirements engineering (E. Yu, 1997), system design methodologies (Castro, Kolp, & Mylopoulos, 2001; Maiden et al., 2004), and security analysis (Elahi & E. Yu, 2007; Liu et al., 2003). In such existing work using i*, the core syntax of the i* framework has often been modified, or has evolved in different directions. Furthermore, experiences teaching

This chapter is an expansion of the following papers/reports:

Horkoff, J., Elahi, G., Abdulhadi, Samer, & Yu, E. (2008). Reflective Analysis of the Syntax and Semantics of the i Framework. Lecture Notes in Computer Science, 5232, 249-260.*

Horkoff, J., & Yu, E. (2010). Finding Solutions in Goal Models: An Interactive Backward Reasoning Approach. Conceptual Modeling, ER 2010 29th International Conference on Conceptual Modeling Vancouver Bc Canada November 14 2010 Proceedings (Vol. 6412, p. 59). Springer-Verlag New York Inc.

Grau, G., Horkoff, J., Schmitz, D., Abdulhadi, Samer, & Yu, E. S. K. (2008). Fostering Investigation, Collaboration, and Evaluation: the i Wiki Experience. iStar (pp. 33-36).*

Grau, G., Horkoff, J., Yu, E., & Abdulhadi, S. (2010). IStar Guide. Retrieved from http://istar.rwth--aachen.de/tiki--index.php?page_re the i* Framework to students has shown that they often modify the syntax presented to them, either intentionally or unintentionally.

In defining interactive analysis procedures for the i* framework, we must consider which version of the syntax to support, including which modifications to allow analysis over. Work in (Horkoff, 2006), included and adapted as part of the current framework, supports analysis over the framework as defined in (E. Yu, 1995; 1997). As we would like our analysis framework to be as generally applicable as possible, we examine common variations from this syntax, in order to assess whether these variations are reasonable, and to ensure that our analysis supports such variations. Although many uses of i* in research have extended the syntax and semantics of the framework, adding additional intentions or notations, we chose to focus our analysis framework on variations of the "core" i* syntax, as defined in (E. Yu, 1995; 1997). Future work can consider how analysis could be expanded to support extra notation or information such as security vulnerabilities (Elahi & E. Yu, 2007) or goal priorities (Amyot et al., 2010).

To determine common i* syntax variation, we survey various instantiations of i* models and compare these instances to the version of i* currently used at the University of Toronto (U of T), based on (E. Yu, 1995; 1997). A reflective analysis of the i* Framework is performed, looking critically at use of the framework, and questioning the assumptions underlying varying syntactic choices. We are interested in discovering the most commonly occurred variations for both students, learning i*, and researchers, applying i* in their work. To this end, 15 student assignments and 15 academic works containing examples of i* models have been surveyed. A qualitative analysis has been performed in order to understand the motivations behind the syntax variations. In our analysis, we compare the perceived motivations behind these variations to the original motivations behind the U of T syntax. As a result of our analysis, we clarify the meaning of several commonly occurring syntactical structures and make recommendations for "strict" and "loose" versions of i* syntax.

Although i* syntax has allowed for flexibility, the need for at least partially automated reasoning in early RE calls for a more precise definition of framework concepts (R7 in Figure 15, below). There is a need to remove ambiguity in certain constructs and relationships in order to allow for automatic propagation in most areas. A more formal definition of i* syntax allows us to define analysis over i* models in a more precise and concise way than was done in (Horkoff, 2006). To this end, this chapter aims to provide a more formal definition of the i* framework than was provided in (E. Yu, 1995; 1997). We use the discoveries in our survey of i* syntax variations to determine how broadly or how flexible to make this more formal definition. We aim to create a balance between removing potential ambiguities and allowing the flexibility for high-level, social concepts, while supporting as many reasonable syntax variations as possible.

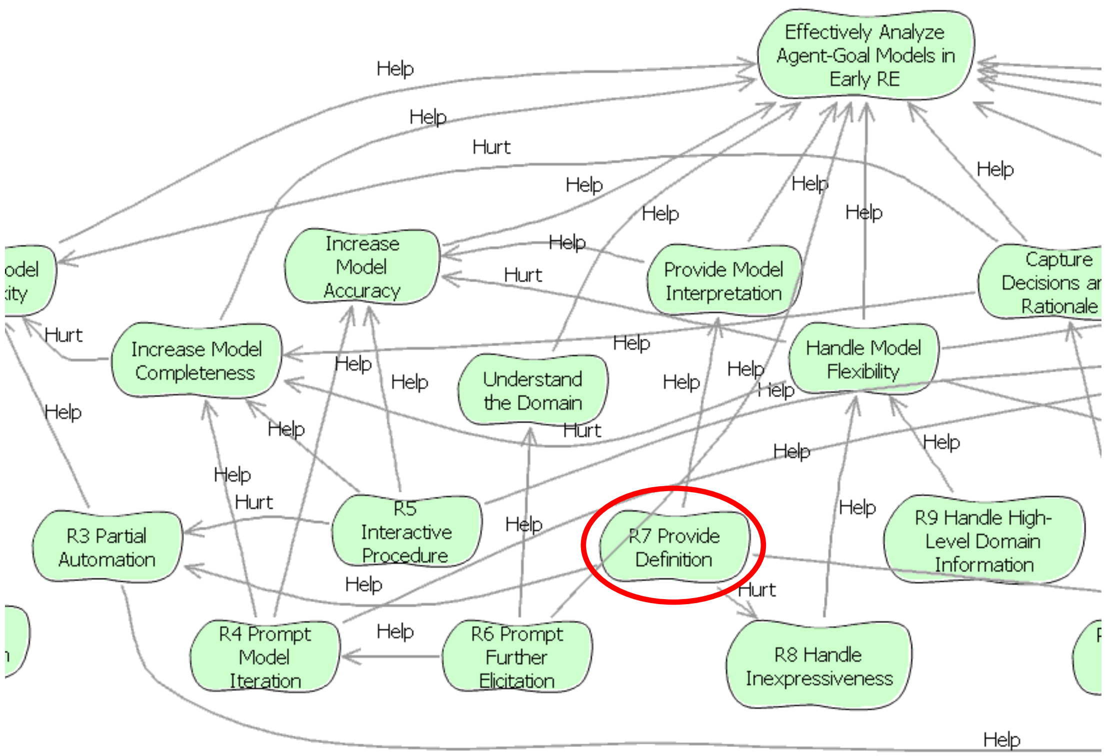


Figure 15: Focus of Current Chapter in the Requirements for analysis of Agent-Goal Models in Early RE

5.1 Background: The i* Framework

In order to create a more formal definition of i*, including a coverage of common variations, a more detailed understanding of the i* syntax is needed. i* models are intended to facilitate exploration of the system domain with an emphasis on social aspects by providing a graphical depiction of system actors including their intentions, dependencies, responsibilities, alternatives and vulnerabilities (Yu, 1997).

The social aspect of i* is represented by *actors*, including *agents* and *roles*, and the associations between them, (is-a, part-of, plays, covers, occupies, instantiates), which can be represented in an *Actor Association (AA) Model*. An *agent* is used to represent a real entity such as a person or system, while a *role* is used to represent a set of responsibilities and intentions taken on by an agent. The *plays* association link is used to represent an agent playing a role, with both the agent and role retaining intentionality independent of particular associations.

Actors depend upon each other for the accomplishment of *tasks*, the provision of *resources*, the satisfaction of *goals* and *softgoals*. *Softgoals* are goals without clear-cut criteria for satisfaction, therefore a softgoal is satisfied when it is judged to be sufficiently satisfied. Agents, associations, and dependencies are used together to create *Strategic Dependency (SD) Models*. In an SD model, a dependency relationship consists of an actor who depends on an intention, the *dependum*, the intention being depended upon, and an actor who is depended upon.

In *Strategic Rationale (SR) Models*, the intentions that motivate the dependencies are explored. Actors are "opened-up" using *actor boundaries* containing the goals, softgoals, tasks, and resources explicitly desired by the actors. Dependency relationships now show the *depender*, the intention depending on another intention, and the *dependee*, the intention being depended upon. The interrelationships between intentions inside an actor are depicted via three types of links. *Decomposition* links show the intentions which are necessary in order to accomplish a task. *Means-Ends* links show the alternative tasks which can accomplish a goal. *Contribution* links show the effects of softgoals, goals, and tasks on softgoals. Positive/negative contributions representing evidence which is sufficient enough to satisfice/deny a softgoal are represented by *Make*/*Break* links, respectively. Contributions with positive/negative evidence that is not in itself sufficient enough to satisfice/deny a softgoal are represented by *Help*/*Hurt* links. Positive/negative evidence of unknown strength can be represented by *Some+*/*Some-* links.

Figure 16 repeats the i* example from Figure 2 and Figure 11, showing a simplified view of the first phase of the youth counseling case study described in the introduction. Figure 17 contains an SD model of the same example. Although these models may seem complex at first, a reader can understand the model by examining it actor by actor and intention by intention. These models contains three actors: the Organization (top), Kids and Youth (bottom left), and Counselors (bottom right). In both models, we can see that the Organization depends on the Counselors to provide the alternative counseling services and for many of its softgoals, for example, High Quality Counseling. Kids and Youth depend on the Organization to provide various counseling services, such as Cyber Café/Portal/Chat Room.

In the SR model (Figure 16) The Organization, an agent, wants to achieve several softgoals, including Helping Kids, Increasing Funds, and providing High Quality Counseling. These goals are difficult to precisely define, yet are critical to the organization. The Organization as the "hard" goal of Providing Online Counseling Services and explores two alternative tasks for this goal: Use Text Messaging and Use Cyber Café/Portal/Chat Room. These alternatives contribute positively or negatively to various degrees to the Organization's goals, which in turn contribute to each other. For example, Use Text Message hurts Immediacy which helps High Quality Counseling.

Both the Counselors and Kids have their own goals to achieve, also receiving contributions from the counseling alternatives. Although the internal goals of each actor may be similar, each actor is autonomous: High Quality Counseling may mean something different for the Counselor than for the Organization.

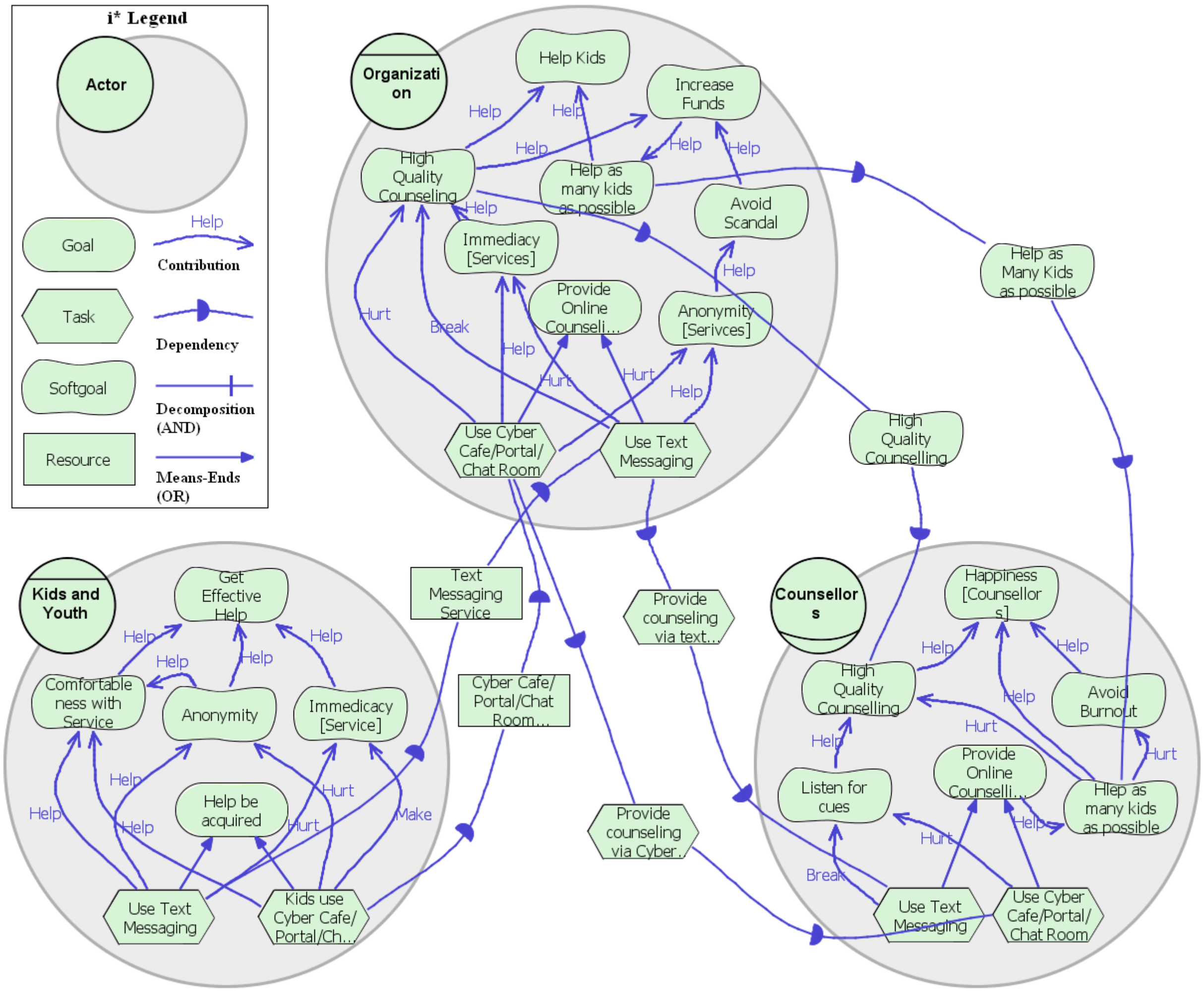


Figure 16: SR Model for Youth Counseling

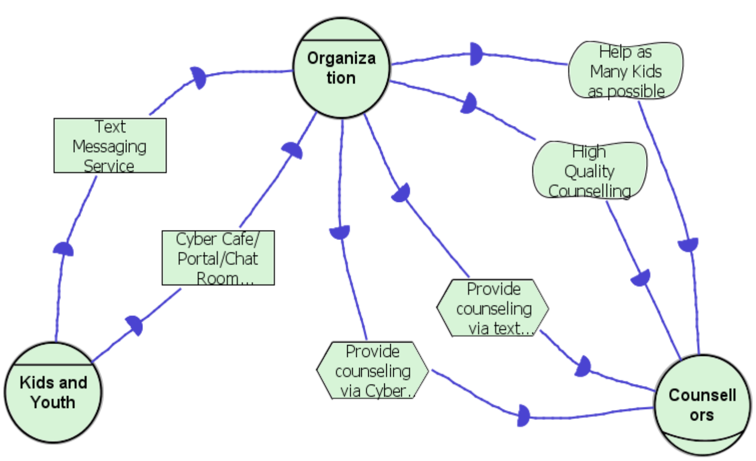


Figure 17: SD Model for Youth Counseling

5.1.1 i* Guidelines on the i* Wiki

In order to enhance the consistency and effectiveness of the modeling process, and provide a resource for students, a set of guidelines based on the U of T i* syntax was developed and added to the i* Guide on the i* Wiki (Grau, Horkoff, E. Yu, & Abdulhadi, 2010). Guidelines were inspired by finding common variations via a survey of assignments in graduate systems analysis courses, as well as a set of available publications, all using i*. The *i** Quick Guide provides a glossary of the *i** constructs and how they should be combined according to their semantics (according to U of T practice) (i* Wiki, 2010). In conjunction with the *i** Quick Guide, the *i** usage guidelines provide assistance for modeling. The *i** quick guide and the guidelines aim to enhance the overall consistency and effectiveness of the *i** modeling processes, reducing variation in practice among users of the *i** modeling framework, and reduced errors for new *i** users.

The *i** usage guidelines are integrated into the glossary of the *i** Quick Guide to help the reader relate between the presented glossary and the associated guidelines. Each guideline deals with a common modeling concept and, in addition to its explanation, provides examples and discussion components, making them more understandable and usable by less experienced *i** users. Each guideline is annotated with initial attributes that indicate the type of guideline (Concept, Naming, Notation, Layout, Methodology, or Evaluation) and the level of guideline difficulty (Beginner, Intermediate, or Advanced). Currently, most of the guidelines are attributed as Beginner. Current attributes, however, could evolve as new and more elaborate guidelines are discovered and added to the *i** Guide.

The *i** Usage Guidelines, are intended to be both an introduction to *i** for new users and a reference guide for experienced users. The guidelines are intended to be flexible recommendations, serving as a catalyst for reflective feedback and future development. To facilitate these objectives, individual wiki pages for all the guidelines are made accessible to all registered *i** wiki users to comment and provide suggestions on individual guidelines. This collaboration aspect fosters an open environment for *i** users and researchers to contribute to charting new and creative ways of presenting, employing, and developing the guidelines.

5.2 i* Variation

We conduct a survey in order to understand common i* syntax variations and their causes. This section describes the method of the survey, provides the survey results, and then categorizes the variations by their perceived motivation, analyzing the causes behind the syntax changes.

5.2.1 Survey Method

Inspired by the findings of the unofficial survey conducted in order to develop the i* Guidelines, 15 essay-type student assignments from 2006 and 2007 and 15 academic papers and presentations were surveyed again, this time taking counts of specific variations to discover the most frequent variations. The assignments were taken from three semesters of graduate courses which covered i* modeling as one of several topics. All assignments were surveyed, unless the students expressed an unwillingness to participate. The papers and presentations were drawn from an introductory roadmap of i* publications (E. Yu, 2011), work appearing in the i* Workshops (see Event page in the i* Wiki, 2010), and work listed in the i* Wiki (see the list of Surveyed Work in Table 71 in Appendix A). The survey covered all constructs of all models in each paper and assignment. The domains covered by the surveyed models were diverse, including health care, banking, and education systems. The inclusion of both academic works and student assignments, allowed for a comparison of the types of variations made by students and newcomers to the variations made in research.

In the surveyed assignments, papers, and presentations, when a model was developed using a convention contrary to the i* guidelines, it was recorded as a variation. If a variation occurred several times in one source, it was only counted once, thus the totals represent the number of assignments or academic works in which the variation occurred. Some variations, such as the use of certain links with certain intentions, were clear-cut to identify, while other variations involved a certain degree of subjective judgment in their identification, for example, deciding that a softgoal should be a goal. In addition to analyzing the variations in 30 sources, we performed a qualitative analysis of the motivations behind the variations, asking questions such as: What did the modeler mean to model with this variation? Is the underlying meaning clear? Was the variation deliberate? Why was the modeler driven to make this variation?

5.2.2 Results

Table 52 lists common variations in the surveyed assignments and papers. It provides total number of variations for each category of variations for student assignments and academic papers or presentations. Although we analyzed both SD and SR models, we detected variations only in SR models. Several of the variations are explained in more detail in the next section.

Table 52: Summary of Common Variations

Category	Variations and grouping categories	Total # of instances for Assignments	Total # of instances for Papers/ Presentations	Total # of instances per variations
Decomposition Links	Decomposition links are drawn directly from goals to tasks	5	4	9
	Decomposition links are used between goals	4	2	6
	Decomposition links are drawn from goals to softgoals	2	3	5
	Decomposition links extend outside actors' boundaries	1	3	4
	Decomposition links are used between Softgoals	2	1	3
	Decomposition links drawn from softgoals to tasks	2	0	2
	Decomposition links are used between resources	1	0	1
	Decomposition links are drawn from goals to resources	0	1	1
Dependency Links	Dependency links are used in more than one strategic relationship	4	4	8
	Softgoal dependency is met by a goal	5	0	5
	Softgoal dependency is met by a task	1	1	2
	Dependency links are used inside actors	0	1	1
	Dependency links do not have dependums	0	1	1
	Dependencies link to actor boundary	0	1	1
Means-Ends Links	Means-Ends links are used between tasks	2	1	3
	Means-Ends links are used between goals	1	2	3
	Means-Ends extend outside actors' boundaries	0	3	3
	Means-Ends are drawn from goals to softgoals	2	0	2
	Means-Ends are drawn from goals to tasks	1	1	2
	Means-Ends are drawn from softgoals to goals	1	1	2
	Mean-Ends are used between softgoals	1	0	1
	Means-Ends are drawn from resources to goals	0	1	1
Contribution Links	Contribution links extend outside actors' boundaries	1	5	6
	Contribution links are drawn from softgoals to tasks	3	1	4
	Contribution links are drawn from Softgoals to goals	1	1	2
	Contribution links are used between goals	1	0	1
	Contribution links are drawn from resources to tasks	1	0	1
Element Types	Softgoal should be goal	10	0	10
	Goal should be softgoal	11	4	15
	Task should be softgoal	7	1	8
	Softgoal should be task	7	0	7
Other	Association links are used between incorrect specialized actors	5	0	5
	Softgoals are not decomposed	2	0	2
	Actors are included inside another actor	0	1	1
	Evaluation Labels are not propagated throughout the model	0	1	1
	Totals	**84**	**45**	**129**

5.2.3 Analysis and Discussion

In this section we select several of the variations to discuss in more detail. We group these variations together under heading representing their perceived causes, for example, "the nature of hard and softgoals" and "means-ends vs. decompositions".

5.2.3.1 The Nature of "Hard" Intentions and Softgoals

Included variations and counts (#):

- Decomposition links are used between Softgoals (3)
- Decomposition links drawn from softgoals to tasks (2)
- Means-Ends are drawn from goals to softgoals (2)
- Mean-Ends are used between softgoals (1)
- Softgoal dependency is met by a goal (5)
- Softgoal dependency is met by a task (2)
- Contribution links are drawn from softgoals to tasks (4)
- Contribution links are drawn from Softgoals to goals (2)
- Contribution links are used between goals (1)
- Contribution links are drawn from resources to tasks (1)
- Decomposition links are drawn from goals to softgoals (5)
- Means-Ends are drawn from softgoals to goals (2)
- Softgoal should be goal (10)
- Goal should be softgoal (15)
- Task should be softgoal (8)
- Softgoal should be task (7)
- **Total: 70**

Many of the variations can be attributed to a misunderstanding of the nature of hard and softgoals. Generally, users confuse hard and soft intentions. Several variations involved having a "hard", non-softgoal intention as a recipient of contribution links. In i*, a goal, task or resource is typically considered similarly to a functional requirement, they are concrete states, actions or entities, respectively. From this point of view, it does not make sense to say that another intention can provide a qualitative contribution to these intentions (either partial or sufficient). To keep the differences between hard and soft intentions clear, the U of T syntax decomposes hard intentions using only AND/OR type links (Decomposition and Means-Ends) in order to ascribe clearly defined decompositions to concrete intentions.

We can also see that modelers occasionally use links associated with hard intentions with softgoals, and that softgoals depend on hard intentions in dependencies. For the first case, as the nature of a softgoal implies qualitative, "good enough" analysis, it is unlikely to be decomposable into strict AND or OR relationships, such as Means-Ends or Decomposition. Although the i* Framework does retain the use of AND and OR contribution links for softgoals, (adopted from the NFR framework), their use is infrequent.

Similarly, when a softgoal dependency is met by a hard intention, this may indicate a problem with the understanding of softgoals. In this situation, if the functional intention (hard intention) is satisfied, the qualitative aspect will also be satisfied. In some cases, the underlying meaning of this type of syntax may be desirable, similar to the situation where a Make link is used from hard intention to a softgoal. However, if the contribution is only partial, or not positive, this syntax should be avoided.

In several cases, modelers have decomposed goals to softgoals, violating the restrictions that goals should only be decomposed to tasks. The nature of hard goals and softgoals implies that a softgoal should not be a means to a hard goal; sufficiently accomplishing a qualitative goal, should not allow the accomplishment of a concretely defined state of the domain. However, we can observe that in i* syntax a softgoal is allowed to be a decomposition element of a task. This seems to contradict the notion of a task as a concrete series of actions. In fact, when this type of syntax is used, we interpret the task to represent not only the concrete actions, but also the desired qualities that this particular task should accomplish in order to be satisfied. For example, in the left snippet of Figure 18, Send Message is only satisfied if the Message is Sent Securely. If the message is sent, but it was not secure, send message is denied. Such a situation can also be created when tasks or goals depend on softgoals.

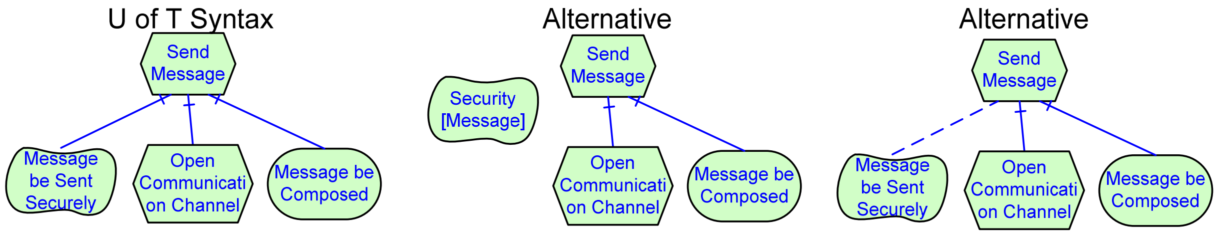


Figure 18: Example Task Decomposition (left), Alternative Syntax Examples (middle, right)

This situation may lead to potential confusion if a modeler or a model reader is not aware of this interpretation, and instead interprets Send Message as the binary, concrete act of sending a message, where, even if the message is not sent securely, it can still be sent. Furthermore, as Send Message becomes a decomposition intention of other functional intentions, this implied qualitative aspect is passed up the decomposition tree to other intentions which could be interpreted as entirely functional. In addition, if a task can be decomposed to a softgoal, why can a goal not also be decomposed in the same way?

Possible Responses. Although a solution to these issues may be to discontinue the decomposition of Tasks to softgoals, there remains a need to explicitly associate non-functional qualities with functional intentions. In the NFR Framework, this was done using a type and topic style of naming, where goals were named by the type of softgoal (security, ease of use, etc.) and their domain specific topic, as "Type [Topic]", see the middle of Figure 18. Alternatively, a visual way to associate softgoals to functional intentions which does not directly affect the evaluation of the functional intentions could be devised, allowing the "hard" intentions would retain their binary meaning. This alternative is shown on the right of Figure 18.

Evaluation Syntax Support. When accounting for this category of variations in the syntax supported by our early RE analysis framework, we aim to support existing practices. Therefore, intention and link combinations which use contribution links with hard intentions or vice-versa will be supported by analysis. There is a possibility that some of these syntax variations may be detected and modified via use of qualitative analysis, as explored in Section 5.2.6. Concerning the combination of hard and soft concepts through decomposition links, we support the situation on the left side of Figure 18, even though it may cause confusions, as it is a commonly accepted practice in i* modeling. The situation in the center of Figure 18 associates the softgoal to the task decomposition only by the semantic content of the intentions. Although the automated portion of the analysis procedure will certainly provide results in this case, it will not automatically associate the softgoal with the task. This is left up to the knowledge of the evaluator, as is the current i* practice. The third situation in Figure 18, although potentially helpful in distinguishing between hard and soft intentions, is not currently part of i* syntax. Therefore, in order to avoid introducing further complexities to i* modeling and evaluation we will not support this syntax in the current version of the framework. Future work can expand i* syntax and the syntax supported by the interactive analysis procedures as appropriate.

5.2.3.2 Means-Ends vs. Decomposition

Included variations and counts (#):

- Decomposition links are drawn directly from goals to tasks (9)
- Decomposition links are used between goals (6)
- Means-Ends links are used between tasks (3)
- Means-Ends links are used between goals (3)
- Decomposition links are drawn from goals to resources (1)
- Means-Ends are drawn from goals to tasks (2)
- **Total: 24**

In the U of T style of i* syntax, deliberate restrictions have been placed on the use of Decomposition and Means-Ends links between elements. A Decomposition link (AND Decomposition) is intended to be used only to decompose tasks into a combination of any element types, where as a Means-Ends link (OR Decomposition) is intended to be used only to refine a goal into alternative tasks. Survey results show that many i* users either chose to ignore or misunderstand these restrictions.

The restrictions concerning Decomposition and Means-Ends links can be justified by the notion of tasks versus goals, and by the desire to prompt for the discovery of alternatives. In (E. Yu, 1997), a goal, by definition, can be accomplished in different ways, whereas a task specifies one particular way of accomplishing something. Thus, in Figure 19, modeling Appointment Be Scheduled as a Goal would indicate that there are several different ways to schedule an appointment, while choosing to model Financial Management as a Task indicates that this refers to one particular way of performing financial management.

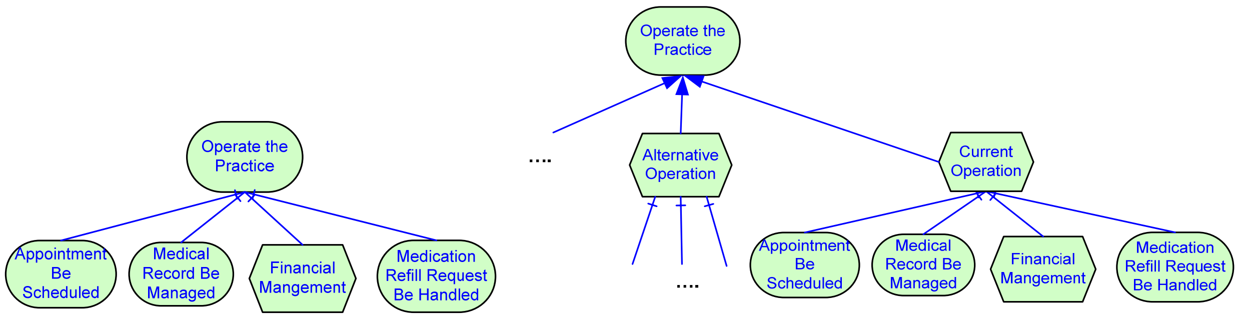


Figure 19: Example of Decomposition Variation Recreated from (Samavi, E. Yu, & Topaloglou, 2008) (left), Redrawn in the U of T Style (right)

In the U of T style, the left side of Figure 19 would be redrawn as shown on the right. By adding the extra task (Current Operation) between the goal decompositions and the original goal, we emphasize that this set of decompositions composes only a single way to decompose and accomplish the task. There can, in fact, be several ways to decompose the high-level goal, each having potentially different effects on qualitative aspects represented as softgoals. Despite these reasons, for reasons relating to the scalability and simplicity of i* models, users often relax the rules concerning Means-Ends and Decompositions.

Possible responses. We propose two levels of i* syntax, a strict level which follows the syntax laid out in Section 5.1 and a looser level which uses syntactical shortcuts. In the strict version of the syntax, restrictions such as those concerning Means-Ends and Decomposition would apply. In the looser level, these restrictions can be relaxed, allowing users to be more concise. Therefore we can consider the left side of Figure 19 as a "shortcut" for the right side. When a modeler chooses to use this simplified syntax, the underlying meaning, represented by the more detailed syntax, should be clear to the modeler and the model readers. If there is any doubt concerning the clarity of meaning, the stricter syntax should be used. More details concerning the rules included in the strict and loose syntax version of i* are included in Section 5.3.

Evaluation Syntax Support. As variations concerning means-ends and decomposition links are common, the evaluation procedures will support means-ends and dependencies decomposed from tasks or goals.

5.2.3.3 Actor Boundaries

Included variations and counts (#):

- Decomposition links extend outside actors' boundaries (4)
- Dependency links are used inside actors (1)
- Means-Ends extend outside actors' boundaries (3)
- Contribution links extend outside actors' boundaries (6)
- **Total: 14**

One frequently observed variation is that a decomposition, means-end, or contribution link extends outside of an actor's boundary. In the U of T version of i* syntax, all of these instances would be replaced by dependency links. It is important to limit non-dependency links to inside boundary of the actors to emphasize on actors' autonomy. In this way, externally visible actor relationships are limited to dependency links, and other actors do not have knowledge of the inside motivations of an actor. This situation better reflects the autonomy of actors occurring in the domain.

By only using dependency links across actor boundaries, one can ensure that the SR model is consistent with the SD model, and avoid confusion translated between the two. However, practitioners frequently violate these rules, and according to detail analysis of the models, scalability and usability issues lead to these variations. Although these variations are not compatible with the notion of actor autonomy, they communicate the same semantics represented with strict rules with a looser syntax which works as a shortcut. For example, Figure 19 shows an example of a variation from (Gans et al., 2005) and its representation using the strict syntax of U of T style.

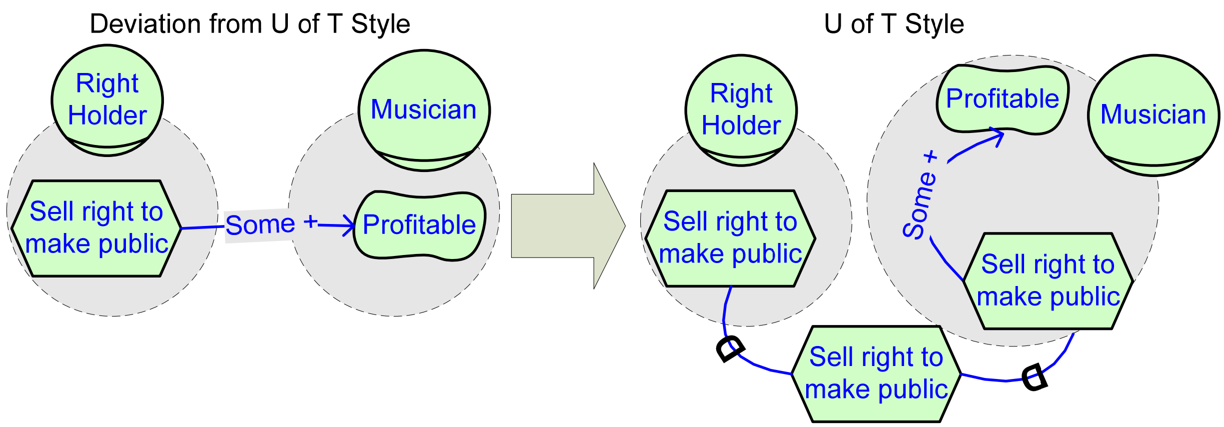


Figure 20 An example variation and its representation using the U of T style.

Possible responses. We include this group of syntax variations in our classifications of loose vs. strict i* syntax. Strict syntax would require dependency links to be the only type of link allowed outside of actor boundaries, while looser syntax would allow other types of links to cross boundaries.

Evaluation Syntax Support. As this type of syntax variation is common, producing models which are remain semantically sensible, the analysis procedures created as part of this work will support such syntax variations. In fact, the automated portion of the analysis procedures described in Chapter 6 and Chapter 7 currently does not take into account actor boundaries, but only intentions, analysis labels, and links. More detail can be found in future chapters.

5.2.4 Variation Results Discussion

By analyzing the results in Table 52, we can note several differences between the variations found in student assignments and academic work. It appears that students have more difficulty in understanding the nature of softgoals, and the differences between soft and hard elements. Although these notions are likely familiar to researchers, students are likely to be new to these ideas. Similarly, we see that students are more likely to have incomplete models, lacking softgoal decomposition, and are more likely to misuse association links. These issues can be addressed by placing greater emphasis on these concepts when teaching i* to new users.

On the other hand, we can observe that researchers are more likely to use non-dependency links outside of actor boundaries. We can postulate that researchers are more likely to adapt the Framework as they see fit. If they are faced with scalability issues, are more likely to deviate from the syntax laid out in (E. Yu, 1995; 1997). No other significant differences between student and research results are found.

Tools to support i* modeling should keep in mind the varying needs of users, including those learning the syntax and requiring guidance, and experienced modelers who understand the notation and may take liberties for expressiveness. The analysis framework produced in this work will support both strict i* syntax as well as reasonable syntactic shortcuts taken as part of the looser syntax, outlined in Section 5.3. Guidance concerning i* syntax will be provided via tool support, described in Chapter 11.

5.2.5 Threats to Survey Study Validity

We can consider several threats to the validity of this study. First, the selection of academic papers and presentations was not performed in a completely random manner, and the surveyor was less interested in papers which did not have deviations. Therefore the selection is not necessarily representative of all research applying i*. However, the presence of a variety of domains in the research papers and assignments indicates that the discovered trends generalize across modeling subject matters.

In addition, only a small number of sources (15 student assignments and 15 research papers or presentations) were reviewed to produce survey results. Recent work by other authors has performed a more extensive review of i* syntax variations. For example, work in (Cares & Franch, 2011) reviewed 146 papers using i*, collecting statistics on how many papers added, removed, or changed syntax representing actors, elements and links. Although this survey had a broader coverage than the survey presented in this chapter, it did not go into a detailed analysis of the specific types of variations, or the perceived motivations for the variations and did not compare use of the language by students to use by researchers.

When analyzing the differences between students and researchers, we see that the student assignments were often longer than the academic works, had more i* examples, and therefore had a higher chance of containing variations. However, as our observations of higher numbers of variations for student assignment is not universal across all counts, the trends observed likely remain valid.

Finally, the qualitative analysis of the variations found in both types of work was performed by the authors of this work, all of whom are very familiar with the U of T style of i* syntax, and who are biased by the flavor of i* which we have learned and used. Therefore, it is possible that the intention behind variations were misinterpreted in some cases. However, if the semantic intention of syntax variations can be misinterpreted, they may be ambiguous and problematic in general.

5.2.6 Utility of Qualitative, Interactive Analysis in Detecting Variations

Work in (Horkoff, 2006) has explored the types of syntax changes prompted by interactive forward analysis. Several of these syntax changes, including “dependency links are used in more than one strategic relationship”, “softgoal dependency is met by a task/goal”, “softgoals are not decomposed”, and “<intention> should be <intention>” are listed as sample i* variations.

For example, consider “softgoal should be goal” or “goal should be softgoal”. (Horkoff, 2006) claims that the use of qualitative analysis labels can help users to understand the differences between hard and soft intentions, as use of partial and full qualitative evaluation labels helps users to reconsider the nature of hard and soft intentions. As a hard intention is typically meant to be binary in nature, application of partial qualitative labels is likely to raise questions in the mind of the modeler. Similarly, when softgoals are automatically always fully satisfied or denied, this contradicts their qualitative “good enough” nature. Qualitative evaluation results can help users to question their selections between softgoals and hard intentions, based on the full or partial nature of the resulting qualitative label.

In Chapter 12, we design and apply empirical studies to discover the types and number of model changes prompted by application of interactive analysis.

5.3 Strict vs. Loose i* Syntax

As discussed in Section 5.2, some variations from the U of T style of syntax can be considered syntactic shortcuts, with approximately the same meaning as the long form (e.g., contribution links across actor boundary). Other variations are not consistent with the semantics of the modeling language (e.g., actor inside of an actor). We can classify the variations as strict variations, meaning if the syntax is followed closely, these variations are errors, and loose variations, meaning these variations are errors even if a looser form of the syntax is followed. Another, possibly more intuitive way to classify such variations is using a distinction between errors and warnings, as is done in a compiler. In this case, all of the variations which can be made as a shortcut are classified as warnings, where as variations which are problematic even when the syntax is loosened are classified as errors. We use the results from above, along with variations listed on the i* wiki to create a list of error and warning rules for the i* syntax. This list can be found in Table 71 in Appendix B. The rules list has been sorted into four categories of rules: actor rules, association link rules, decomposition rules, and (non association) link rules. The first four columns provides an error summary, description, corresponding link to the i* Wiki Guidelines (if applicable) and the type of rule (error/warning). The last three columns are described more in Section 5.5. Syntax checks for i* are discussed again in Chapter 11, when describing the features of OpenOME.

5.4 A Formal Definition of i*

We introduced a more formal description of the i* syntax described in Section 5.1. Our aim was to help remove ambiguity in the syntax and support semi-automated analysis, as described in the introduction to the current chapter.

In our description, we use the following notation:

- $\mapsto$ is used as a mapping from an intention or relation to a member of a set, so $i \mapsto \{a, b\}$ means that i maps to either a or b.
- $\rightarrow$ is used to represent relationships between elements, so if $(i_1, i_2) \in \mathcal{R}$ we write this as $\mathcal{R}: i_1 \rightarrow i_2$.

An Agent-Goal (i*) Model. We express agent-goal model concepts such as actors and softgoals formally as follows.

Definition: agent-goal model. *An i* model is a tuple* $\mathcal{M} = <\mathcal{I}, \mathcal{R}, \mathcal{A}>$*, where* $\mathcal{I}$ *is a set of intentions,* $\mathcal{R}$ *is a set of relations between intentions, and* $\mathcal{A}$ *is set of actors.*

Definition: element type. *Each intention maps to one type in the IntentionType set,* $\mathcal{I} \mapsto$ *IntentionType, where IntentionType = {Softgoal, Goal, Task, Resource}.*

Definition: relation type. *Each relations maps to one type in the RelationType set,* $\mathcal{R} \mapsto$ *RelationType, where RelationType =* $\{R^{me}, R^{dec}, R^{dep}, R^{c}\}$*. These relationships correspond to means-ends, decomposition, dependency, and contribution links, respectively.* R^c *can be broken down into a further set ContributionType =* $\{R^m, R^{hlp}, R^u, R^{hrt}, R^b\}$ *where if* $r \in \mathcal{R} \mapsto R^c$ *then* $r \mapsto$ *ContributionType. The contribution link types correspond to make, help, unknown, hurt, and break, respectively.*

Definition: relation behavior. *The following relationships are binary (one intention relates to one intention,* $\mathcal{R}: \mathcal{I} \rightarrow \mathcal{I}$*):* R^{dep} *and* R^c*. The remaining relationships,* R^{me}*,* R^{dec}*, are (n+1)-ary (one to many intentions relate to one intention),* $\mathcal{R}: \mathcal{I} \times \ldots \times \mathcal{I} \rightarrow \mathcal{I}$*. When describing relations, the intentions on the left hand side of a relation are referred to as sources (s), while the intention on the right hand side is referred to as a destination (d).*

The formalism could be supplemented to include actor types and association links. Currently, these types do not play a role in the automated portion of the analysis procedures described in Chapter 6 and Chapter 7. We leave their inclusion in the formalism to future work.

Some+ and Some- links, as described in Section 5.1, could be included in the formalism as additional R ∈ *ContributionType.* However, the analysis procedures described in Chapter 6 and Chapter 7 do not differentiate between these links and Help and Hurt, respectively. In other words, the procedure conservatively treats Some+ as Help and Some- as Hurt. We exclude these links from *ContributionType* in the formalism described above for the sake of simplicity. More details concerning the treatment of these links in analysis can be found in Chapter 6.

5.4.1 Useful Concepts

We define several other useful concepts such as leaves, roots, positive and negative links.

Definition: leaf or root intention. *An intention* $i \in \mathcal{I}$ *is a leaf if there does not exist any relation,* $r \in \mathcal{R}$ *such that* $r : \mathcal{I} \rightarrow i$ *or* $r : \mathcal{I} \times \ldots \times \mathcal{I} \rightarrow i$*, it is a root if there does not exist any relation,* $r \in \mathcal{R}$ *such that* $r : i \rightarrow \mathcal{I}$ *or* $r : i \times \ldots \times \mathcal{I} \rightarrow \mathcal{I}$.

Definition: positive or negative link. *A relation* $r \in R$ *is positive if* $r \longmapsto Pos = \{R^m, R^{hlp}\}$*, it is negative if* $r \longmapsto Neg = \{R^b, R^{hrt}\}$.

5.5 Variation Coverage

When providing interactive analysis procedures for the i* framework, we must determine over which syntax to support analysis. We have developed the formal definition of i* in the proceeding section, describing the i* syntax supported by the analysis procedures in Chapter 6 and Chapter 7. Our aim was to describe a syntax which both removed ambiguity and allowed for flexibility in syntax. Section 5.3 has described two levels of i* syntax: strict and loose (errors and warnings). The formalization of i* provided in the previous section is intended to allow for looser (warning) syntax variations, while not allowing for stricter (error) variations. The fifth column in Table 71 describes whether or not the formalism defined supports a variation, restricts from creating a variation, or does not specify whether a variation is allowed. For example, means-ends links between two tasks is allowed in the loose syntax (is a warning), and is supported by the formalism. Dependency links from or to an actor boundary has not been defined as a permissible connection for dependency links and is not supported by the formalism. Here, we assume that the description of link behavior is complete, in other words, if links between certain elements have not been described in our formalism, they are not permitted. Having an actor within an actor is an erroneous variation, but our formalism does not yet specify whether or not this behavior is allowed. Generally, the table shows that the formalism is very permissive; it allows all variations which would produce warnings. The formalism either does not support or does not yet specify syntax which would produce errors. For example, it does not say anything about actor containment, and therefore does not specify whether errors related to this would be avoided.

5.6 Conclusions

As the i* Framework has been adapted and expanded for use in several contexts, analysis procedures over i* must be specific concerning the supported syntax. In this chapter we have reviewed the "core", U of T style syntax provided in (Yu, 1997). We have performed a survey, collecting and analyzing common variations in i* syntax. Results of this survey have lead to the definition of loose and strict levels of i* syntax, dividing common variations into errors and warnings. Using the description of i* in (Yu, 1997), and taking into account variations, a more formal definition of core i* syntax was provided, including useful concepts such as roots and leaves. This formalism was designed to be inclusive, allowing modelers to draw models including common variations in the looser i* syntax (warnings). In this way, the procedures in the following chapters which use this formalism can be more generally applied to a variety i* model variations.

Returning to the requirements for the analysis of agent-goal models in early RE, as outlined in Chapter 3, this chapter has provided a definition of the i* Framework, addressing R7. In doing so, we have aimed to be as flexible as possible, making a positive contribution towards handling the inexpressiveness required for model flexibility in early RE (R8). We summarize the contributions of this chapter in Figure 21.

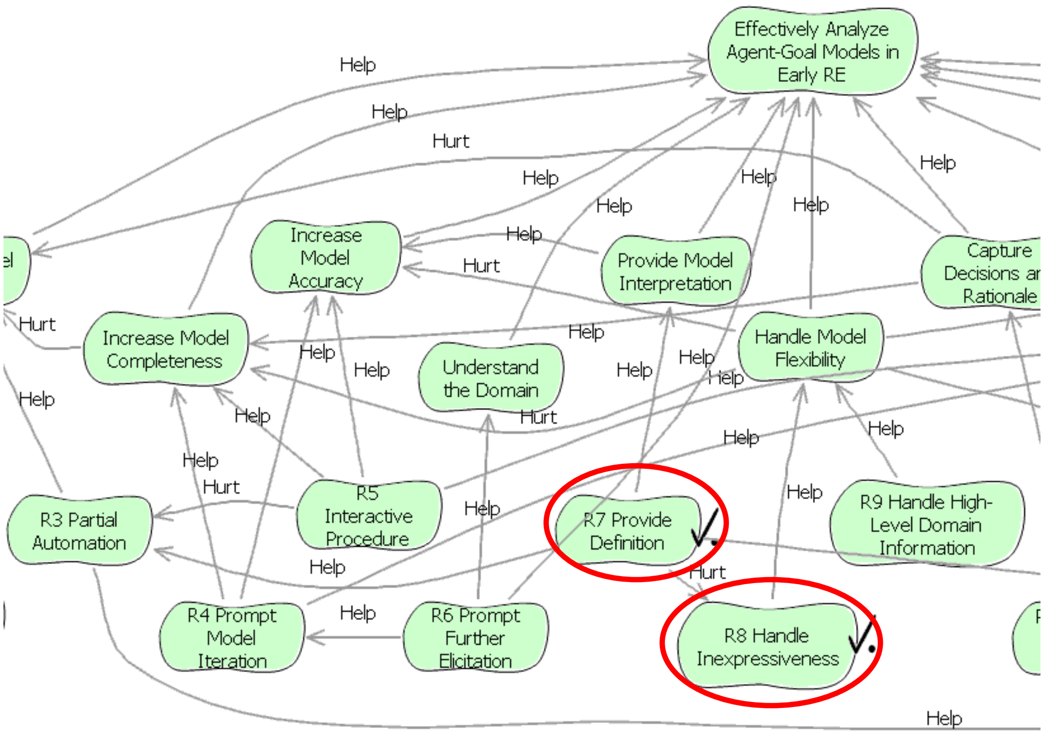


Figure 21 Contributions of Chapter 5 to the Requirements for analysis of Agent-Goal Models in Early RE

Chapter 6 Iterative, Interactive, Forward Satisfaction Analysis of Agent-Goal Models

In this Chapter we introduce a qualitative, interactive evaluation procedure for goal- and agent-oriented models, allowing the user to compare alternatives in the domain, asking "what if?" type questions. The procedure was first introduced in (Horkoff, 2006), but is described here using the formalism from Chapter 5. Examples of procedure use are provided using sample models from the Counseling Service and Trusted Computing Case studies, summarized in Chapter 12.

The procedure reintroduced in this chapter addresses several of the requirements for analysis of agent-goal models in early RE described in Chapter 3. One of the most prominent goals of this procedure is to facilitate analysis over a domain, allowing for "what if?" analysis questions, addressing R13. In addition, the procedure provides partial automation to handle model complexity (R3), has a running time which is linear to the size of the model (R1), is interactive in an effort to improve model accuracy and completeness (R5), comes with a definition of analysis

This chapter is an expansion of the following papers/reports:

Horkoff, J., & Yu, E. (2009a). Evaluating Goal Achievement in Enterprise Modeling – An Interactive Procedure and Experiences. The Practice of Enterprise Modeling (pp. 145-160). Springer.

Horkoff, J., & Yu, E. (2009b). A Qualitative, Interactive Evaluation Procedure for Goal- and Agent-Oriented Models. CAiSE'09 Forum, Vol-453 (pp. 19-24). CEUR-WS.org.

Horkoff, J., & Yu, E. (2010b). Interactive Analysis of Agent-Goal Models in Enterprise Modeling. International Journal of Information System Modeling and Design (IJISMD). IGI Global.

Horkoff, J., & Yu, E. (2010a). Finding Solutions in Goal Models: An Interactive Backward Reasoning Approach. Conceptual Modeling (ER 2010) 29th International Conference on Conceptual Modeling, Vol. 6412, p. 59. Springer-Verlag New York Inc.

constructs (R7), accommodates high-level domain concepts (R9), captures human judgments over the model (R10), and is relatively simple to apply when compared to existing analysis procedures (R15) (see Section 3.3.9 for a consideration of existing procedure usability). The goals addressed by the procedure introduced in this chapter are highlighted in Figure 22.

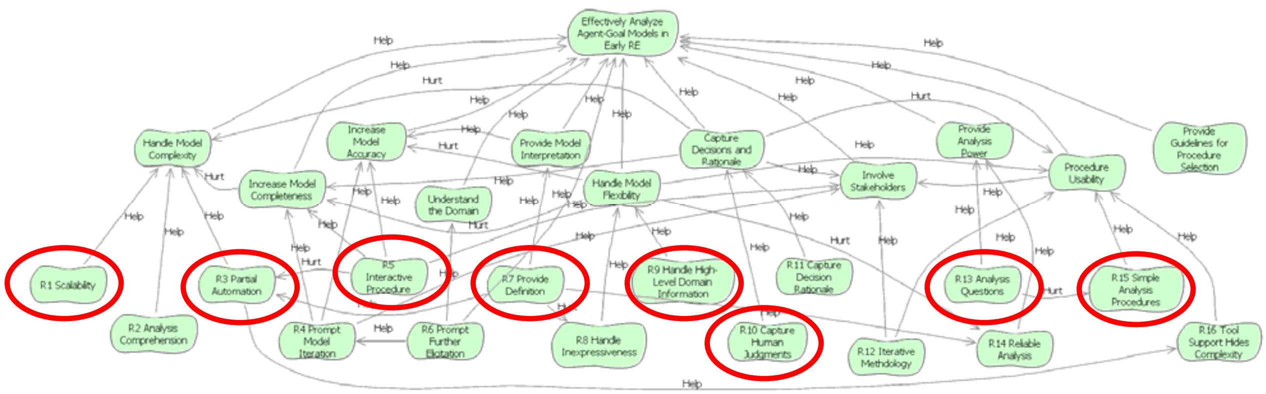


Figure 22: Requirements for analysis of Agent-Goal Models in Early RE Addressed by this Chapter

6.1 Analyzing Agent-Goal Models Manually

In order to further motivate the need for systematic analysis over goal models, including at least partial automation of the procedure, we attempt to perform forward analysis on several example models manually. In our first example, we return to the model from Figure 1, Figure 11, and Figure 16, repeated below in Figure 23. In Section 1.1.1 we asked "Which counseling alternative is the most effective?". We could start this analysis by considering the alternative where Use Text Messaging, represented as a task in the model, is implemented, and Use Cyber Café/Portal/Chat Room, another task, is not implemented. The reader can try to use their knowledge of i* syntax provided in Chapter 5 to trace the effects of the satisfaction or denial of these tasks through the links in the model. In one path inside of the Counselor Actor, Provide Online Counseling would be satisfied via one of the alternative tasks, this would result in a positive contribution to Help as many kids as possible, which would help Happiness Counselors, but hurt Avoid Burnout, which would indirectly hurt Happiness Counselors. In another path, Listen for cues is both denied and weakly satisfied, likely being either fully or partially denied, which in turn hurts High Quality Counseling, which would have an indirect negative effect on Happiness Counselors. Considering these effects, is Happiness [Counselors] denied? Partially denied? Conflicted?

When tracing the effects manually, it is cognitively difficult to follow all paths and make these decisions manually. In this example, we have not even left the boundaries of the Counselor. When considering the effects of dependencies into and out from the actor, tracing the effects of alternatives through the paths of links becomes even more complicated.

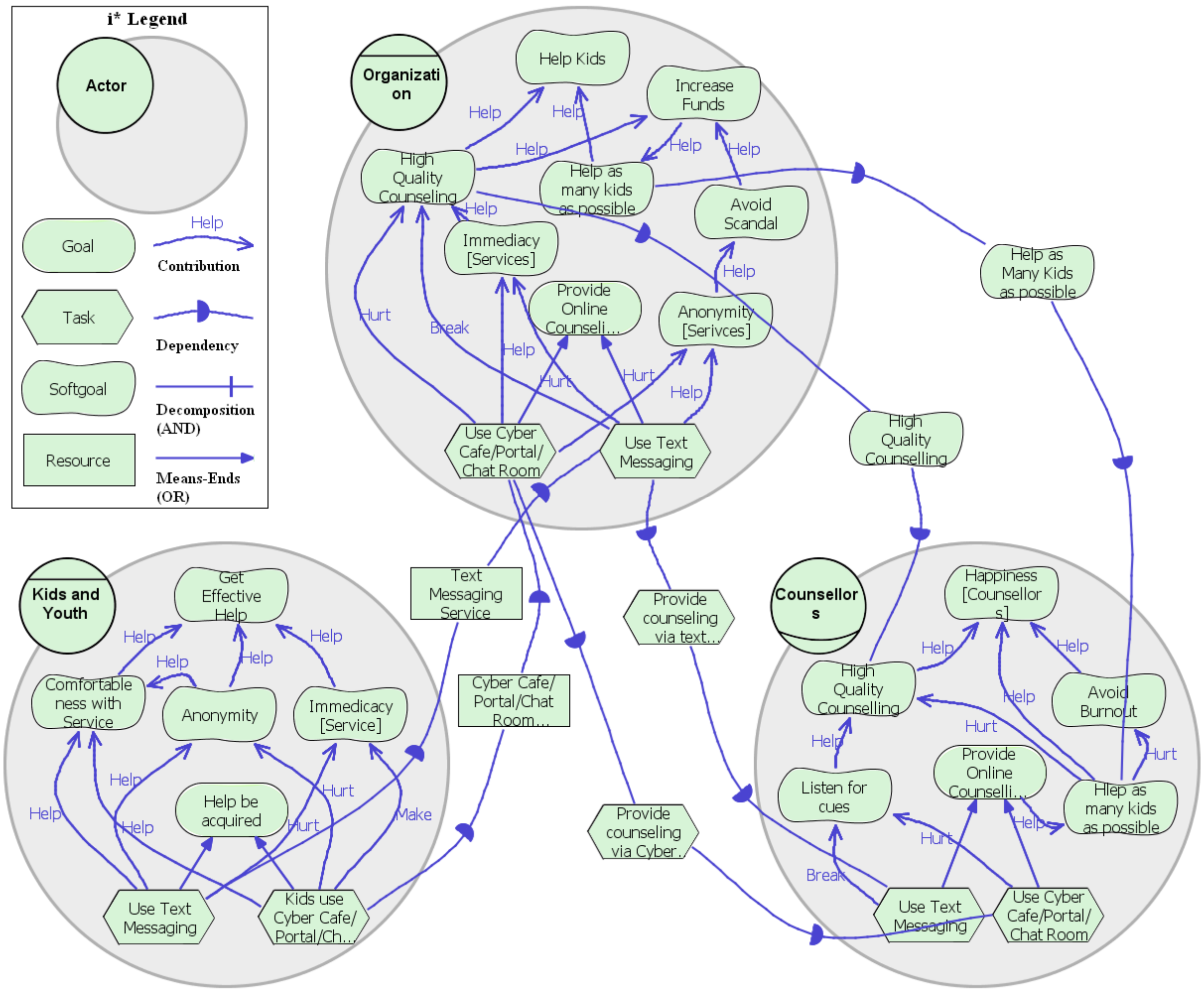


Figure 23: Counseling Service Sample SR Model

We can attempt similar analysis, using another example model concerning software or PC (personal computer) product piracy repeated from (Horkoff, 2006) in Figure 24. Relevant actors in this situation include the PC user, PC product producer and the data pirate. A PC user can acquire PC products legally or acquire them illegally through a data pirate. When acquiring PC products, a PC user may be interested in both saving money, and in abiding by licensing regulations. Legal acquisition helps to abide by licensing regulations, but has a negative effect on the software users desire to find affordable products. Acquiring PC products illegally helps to satisfy the goal of finding affordable products, but breaks licensing regulations.

From a different perspective, the PC product producer wants to sell PC products in order to make a profit. In order to be profitable, the PC product provider must make PC products desirable and also depends on the PC user to abide by licensing regulations. Allowing the use of peer-to-peer technology in PC products can help to make these products more desirable, but can also allow the Data Pirate the ability to make pirated content available.

In performing domain analysis over this model, we can ask: what are the effects of obtaining software or PC products legally or through a pirate? In Figure 24, the PC User can Obtain PC products legally, via the PC Product Provider, or illegally, via the Data Pirate. Each choice has different effects on actor goals. This model serves as an example of the sort of analysis that i* can facilitate. For example, if the PC User Abides by Licensing Regulations, how does this affect the PC Product Provider's ability to make a Profit? We can see that the PC Product Provider depends on the PC User to Abide by Licensing Regulations, and that this softgoal has a positive effect on Profit. As the effects of this choice on Profit are relatively easy to pick out in the model, this is an example of a type of analysis question which can be answered without a systematic procedure. Generally, questions which involve following the effects of one intention on another intention, or an intention on an intention on an intention – a few jumps of separation - can be answered simply by studying the model.

However, some questions prompted by i* models are not straightforward to answer by studying the model. For example, if the PC Product Provider decides to not Allow Peer-to-Peer Technology, what effect will this have on Sell PC Products for Profit? Allow Peer-to-Peer Technology is now denied, producing negative effects for both Desirable PC Products and Profit. However, the Data Pirate depends on Allow-Peer-to-Peer Technology in order to Make Content Available Peer-to-Peer, and the PC User depends on this task for Pirated PC Products in order to obtain PC Products from a Data Pirate. If the PC User is not able to Obtain PC Products from the Data Pirate, it will likely Purchase PC Products. As a result, the Abide by Licensing Regulation softgoal is satisficed, which helps Profit. However, if we recall the previous path through Desirable PC Products, Profit is also negatively affected, resulting in conflicting evidence for Profit.

As the above description shows, manually tracing the affects of alternatives can be complex. We can see that in order to derive a final judgment of satisfaction or denial for an intention we need to be able follow the chain of affects of one intention on another, knowing how to interpret the affects of each type of link. In addition we may need to trace through the affects in multiple paths, if a single intention affects multiple other intentions. Performing this process in an ad hoc manner with no intermediate storage of results is difficult and can be prone to errors and inconsistencies.

We can observe that the model in Figure 24 is relatively simple. What if the same sort of analysis was needed for a more complicated model? These examples make it clear that a systematic evaluation procedure providing partial automation is needed.

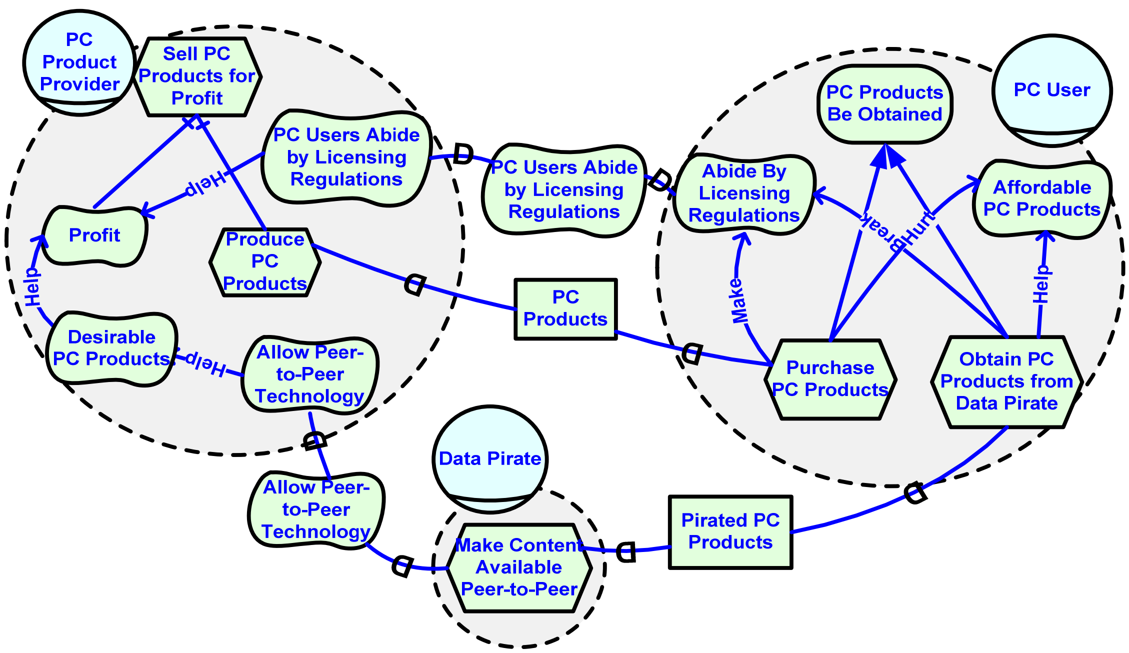


Figure 24: Simplified Example from the Trusted Computing Case Study in (Horkoff, 2006)

6.2 Summary: Modeling and Analysis Suggested Methodology

In order to facilitate the use of agent-goal models for early RE, we provide a set of guidelines for model creation, iteration, and analysis. We outline this methodology in this section, illustrating how forward analysis can be used as part of a modeling process. A more detailed description of this methodology can be found in Chapter 8. The steps of the method are intended to be iterative; in each step new artifacts or changes relevant to previous steps may be discovered, and should be added to the model.

The first steps of the methodology involve identifying the scope or purpose of the modeling, whether it be to elicit requirements, understand the domain, choose between alternatives, etc. Next, model participants (stakeholders) or other sources are identified. In order to start creating the model(s), we recommend first identifying relevant actors and associations, creating an Actor Association model. Then, relevant dependencies between actors can be elicited, created a Strategic Dependency Model. In the next step, actors are "opened" up to create a Strategic Rationale model, where actor intentions are identified and added to the model. Existing dependencies from the SD model are matched to actor intentions. Further relationships between intentions are identified.

The next steps in the suggested methodology focus on the application of analysis. First, analysis is used as a means to check the "sanity" or relative completeness or accuracy of a model. Forward analysis is applied by identifying all of the leaf intentions and trying extreme or borderline alternatives: what if everything is implemented (all alternatives chosen)? What if nothing is implemented? What if likely alternatives are implemented? If analysis results are not sensible (for example, implementing nothing satisfies many goals) then the model should be iterated over. A similar step is performed for backward analysis, identifying all the root intentions and asking: is it possible for all roots to be satisfied? Is it possible for the minimum (domain specific) set of targets for roots to be satisfied? Iteratively, what is the maximum set of roots which can be satisfied? Finally, the methodology recommends using the model to ask relevant domain-driven questions, including aiding in the selection of system alternatives. The suggested methodology is summarized in Figure 25.

Apply the following steps iteratively:

Stage 1: Purpose and Elicitation

- Identify scope or purpose of the modeling process.
- Identify modeling participants and/or model sources.

Stage 2: Model Creation

- Identify relevant actors and associations.
- Identify relevant dependencies.
- Identify actor intentions.
- Identify relationships between intentions.

Stage 3: Analysis

- Alternative Effects (Forward Analysis)
  - Identify all leaf intentions in the model, evaluate:
    - Implementing as much as possible.
    - Implementing as little as possible:
    - Reasonable Implementation Alternatives.
- Achievement Possibilities (Backward Analysis)
  - Identify all roots in the model, evaluate:
    - Maximum targets.
    - Minimum targets.
    - Iteration over minimum targets.
- Domain-Driven Analysis (Mixed)
  - Use the model to answer interesting domain-driven questions.

Figure 25 Summary of Suggested Model Creation and Analysis Methodology

6.3 Background: Forward Qualitative i* Analysis from Horkoff (2006)

As described in Section 2.3.1.2, work in (Horkoff, 2006) introduced a forward, qualitative, and interactive evaluation procedure for the i* framework. This procedure adapted and expanded NFR analysis by considering the frequency of human judgment, exploring the meaning of initial evaluation labels and evaluation alternatives, expanding the procedure to work in i*-specific syntax (dependency links), expanding the ordering of analysis labels, differentiating between unknown and no label, allowing for more interactivity in the evaluation algorithm, specifying more precise pseudocode in the algorithm, and considering propagation over more complex syntax structures, such as a mixture of links, and links to links.

Although this procedure provides a useful basis for forward evaluation in the framework introduced in this work, the procedure was described using prose, examples, and pseudocode. In the next section of work, we expand the description of the procedure using the formalism introduced in Section 5.4. We update the pseudocode to describe the procedure more succinctly. Conditions for procedure termination in the presence of cycles are modified to ensure value convergence. Although work in Horkoff (2006) described the procedure in great detail, it did not specify in detail how the procedure should be used as part of a modeling and analysis methodology. One of the most significant additions to this work made as part of the new framework is a detailed methodology, both describing how analysis fits into model creation (Section 8.1), and how both the forward and backward analysis procedure can be used together to improve the model and analyze the domain (Section 8.2).

6.4 Expanded Description of Forward Analysis

After providing an overview of the procedure, we describe the detailed steps of the evaluation procedure, including an explanation of the required concepts using the formalism defined in Chapter 5. The procedure is illustrated with several example evaluations.

6.4.1 Procedure Overview

The procedure starts with an analysis question of the form "How effective is an alternative with respect to model goals?" The procedure makes use of a set of qualitative evaluation labels assigned to intentions to express their degree of satisfaction or denial. The process starts by assigning labels to intentions related to the analysis question. These values are propagated through the model links using defined rules. The interactive nature of the procedure comes when human judgment is used to combine multiple conflicting or partial values to determine the satisfaction or denial of a softgoal. The final satisfaction and denial values for the intentions 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"), stimulating further analysis and potential model refinement.

6.4.2 Detailed Procedure Steps

We provide a more detailed description of the steps of the evaluation algorithm using both pros and the more precise definitions from Chapter 5. Pseudocode summarizing the implementation is provided in Section 6.4.11.

1. **Initiation:** The evaluator decides on an alternative 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 intention.
>
> 3. **Softgoal Resolution:** Label bags are resolved by applying automatic cases or manual judgments, producing a result label which is added to the label queue.

4. **Analysis:** The final results are examined to find the impact of alternatives on stakeholder goals. Model issues can be discovered, further alternatives are evaluated.

6.4.3 Qualitative Analysis Labels and Predicates

We adopt the qualitative labels used in NFR evaluation and previous examples of i* evaluation, replacing "weakly" with "partially". The resulting labels are *satisfied*, *partially satisfied*, *conflict*, *unknown*, *partially denied*, and *denied*. The *Satisfied* (✓) label represents the presence of evidence which is sufficient to satisfy a goal. *Partially satisfied* (✓•) represents the presence of positive evidence which is not sufficient satisfy a goal. *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. In addition to these labels we introduce the "None" label to indicate a lack of any label. We use partial labels for tasks, resources, and goals, despite their clear-cut nature, to allow for greater expressiveness.

In order to express evaluation labels as part of our formalism we introduce analysis predicates, similar to those used in Tropos Analysis (Giorgini et al., 2005).

Definition: analysis predicates. *We express agent-goal model analysis labels using the following set of predicates,* $\mathcal{V}$*, over* $i \in \mathcal{I}$: $v(i) \in \mathcal{V} \mapsto$ *AnalysisPredicates = {S(i), P S(i), C(i), U(i), PD(i), D(i)} where S(i)/PS(i) represents full/partial satisfaction, C(i) represents conflict, U(i) represents unknown, and D(i)/PD(i) represents full/partial denial.*

For example, we express the initial values in our analysis question over the counseling service model in Section 6.1 as S(Use Text Messaging) and D(Use Cyber Café/Portal/Chat Room). If these predicates are true, Use Text Messaging is satisfied and Use Cyber Café/Portal/Chat Room is denied. If the first predicate is false, Use Text Messaging is not fully satisfied. If the second predicate is false Use Cyber Café/Portal/Chat Room is not fully denied. These predicates tell us nothing about the application of the other evaluation labels to this intention. For example, S(Use Text Messaging) does not imply that D(Use Text Messaging) is false.

6.4.3.1 Conflict Distinctions

In the analysis procedures, we choose to treat conflict labels as a value to propagate, as opposed to something derived from other values. In other words, S(Use Text Messaging) and D(Use Text Messaging) does not imply C(Use Text Messaging). This allows the user greater flexibility, giving the option to resolve conflicts; potentially avoiding situations in complex models where analysis results consist mostly of conflict values (see, for example, Table 47, where many intentions receive a direct, C, or indirect, PS and PD, conflict value as a result of analysis).

We still use the term "conflict" between labels to indicate a situation where more than one analysis predicate holds for an intention and those predicates are conflicting.

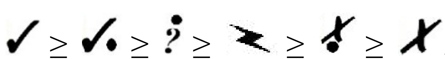
Definition (conflict label vs. conflict between labels). *A conflict label is the label originating when a user has selected conflict in human judgment. A conflict between labels for an intention* i $\in \mathcal{I}$ *is when a predicate from more than one of the following four sets is true:* $\{S(i), PS(i)\}$, $\{U(i)\}$, $\{C(i)\}$, $\{PD(i), D(i)\}$.

6.4.3.2 Analysis Label Order

Unlike the approach in Tropos analysis (Giorgini et al., 2005), we are not able to define a total order over analysis predicates, such that for $v(i) \in \mathcal{V}$, $v_1 \geq v_2 \Leftrightarrow v_1 \rightarrow v_2$, as there are no implication relationships between satisfaction/denial values and unknown values. We chose not to add implication values from {S, PS, PD, D} to Conflict labels (e.g., $PD(i) \wedge PS(i) \rightarrow C(i)$), due to our treatment of such labels as described in the previous section. We are, however, able to define and utilize the following partial orders.

$$\forall i \in I: \quad S(i) \geq PS(i) \Leftrightarrow S(i) \rightarrow PS(i) \qquad (1)$$
$$D(i) \geq PD(i) \Leftrightarrow D(i) \rightarrow PD(i).$$

In addition, we can define a conceptually useful total order where v1 ≥ v2 implies that v1 is more desirable (or "higher") than v2. This order is as follows:

, or (2)

$S(i) \geq PS(i) \geq U(i) \geq C(i) \geq PD(i) \geq D(i)$

Here we chose an optimistic ordering between *U(i)* and *C(i)*, with the idea that no information (unknown) is better than conflicting information.

6.4.4 Initial Analysis Labels

In order to start an evaluation of a model, a set of initial values reflecting an analysis question is placed on the model. For example, in Figure 23, if we wanted to ask our previous analysis question, "What if the Counseling Organization implements Use Text Messaging and not Use Cyber Café/Portal/Chat Room?" we would place initial values as shown in Figure 26 (circled labels). Often, initial values are assigned to leaf intentions in the model, but initial values are not restricted to leaf value. See Section 4.2.2.3 in (Horkoff, 2006) and Section 6.4.11 for a description of the treatment of non-leaf initial values in the analysis algorithm.

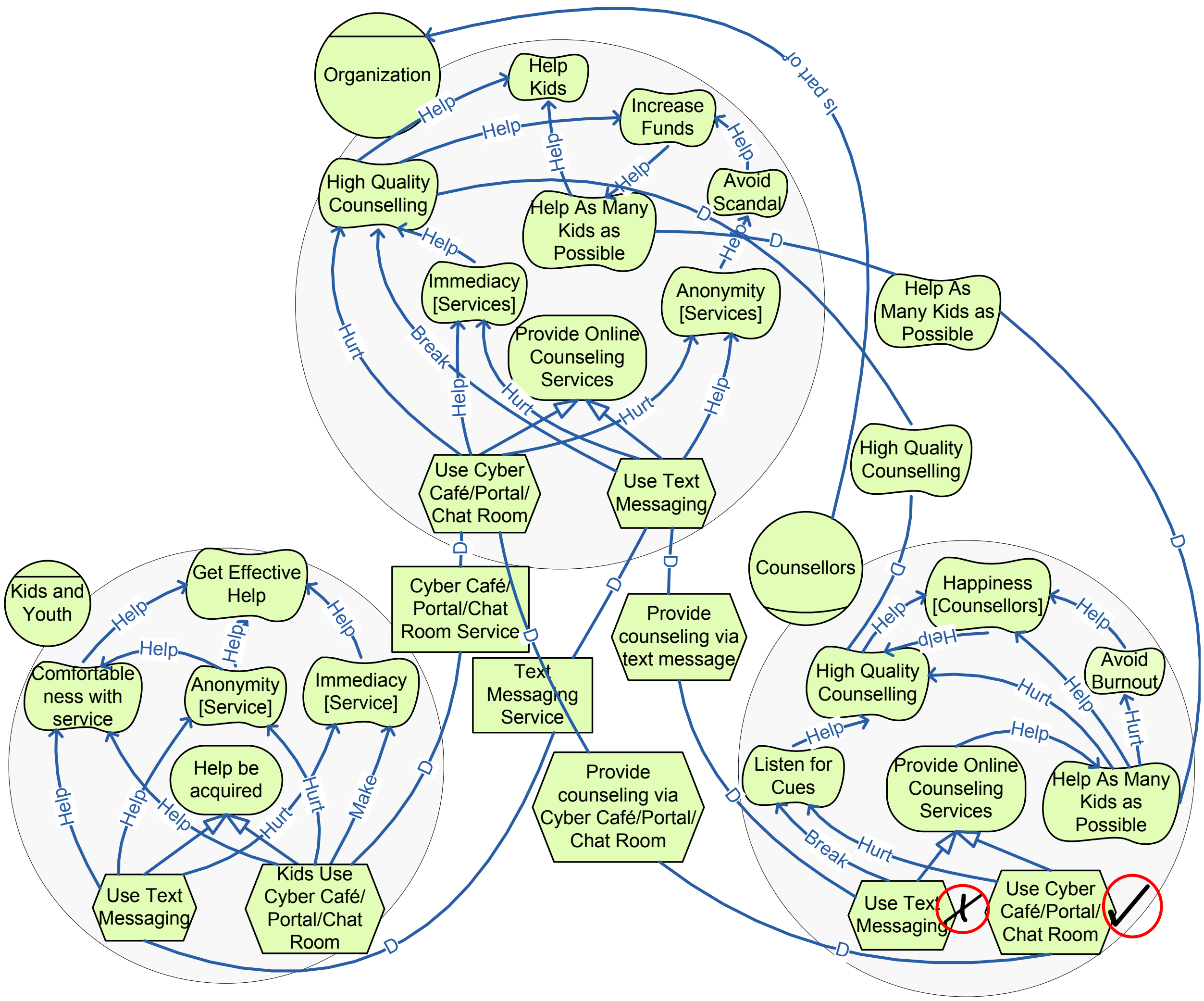


Figure 26: Counseling Service Model showing Example Initial Evaluation Labels

We can express the selection of initial analysis values more precisely as follows:

Definition: initial analysis labels. *For some subset of intentions,* $i_1 \ldots i_n \in \mathcal{I}$, $n \leq |\mathcal{I}|$ *(the number of intentions), within an agent-goal model, a selection of analysis labels,* $v(i_1) \ldots v(i_n) \in \mathcal{V}$. *This selection represents an analysis question over the domain. We refer to the set of initial labels* $v(i_1) \ldots v(i_n)$ *as* $\mathcal{IL}$ *for simplicity.*

6.4.5 Model and Analysis Alternatives

It is useful to define our use of the term "alternative". Often, when referring to i* models, "alternative" is used to mean the choice between means in a means-ends relationship. For example, in Figure 23, the counseling service can implement Use Text Messaging or Use Cyber Café/Portal/Chat Room. In Figure 24, the PC User can Purchase PC Products or Obtain PC Products from the Data Pirate. We will refer to this type of alternative as alternatives for a goal.

Definition: alternatives for a goal. *Alternative means to an end. Each task,* $i_1 \ldots i_n \in \mathcal{I} \longmapsto$ *Task, which can satisfy a goal as part of a means-ends relation,* $R^{me} : i_1 \times \ldots \times i_n \rightarrow i_m$*, where* $i_m \in \mathcal{I} \longmapsto$ *Goal. Note: this can be relaxed in loose syntax to allow for each i to map to any intention type in the IntentionType set.*

Many i* models capture several such alternatives. In addition to this structure, by assigning evaluation labels to alternatives over a goal, we can select a combination of full/partial satisfaction or denial over alternatives for a goal. We can refer to this as an alternative selection for a goal. If, for example, in Purchase PC Products were given a satisfied label and Obtain PC Products from the Data Pirate was given a denied label, this would represent one alternative selection. If, for example, in Figure 24 both alternatives for a goal were given labels of partially satisfied, this would be another alternative selection, where each action is "somewhat" taken, for example, maybe some products are bought legally while others are downloaded from a peer-to-peer network.

Definition: alternative selection for a goal. *An assignment of analysis labels to alternatives for a goal. Each task,* $i_1 \ldots i_n \in \mathcal{I} \longmapsto$ *Task, which can satisfy a goal as part of a means-ends relation,* $R^{me} : i_1 \times \ldots \times i_n \rightarrow i_m$*, is assigned an analysis value* $v(i_1) \ldots v(i_n) \in \mathcal{V}$.

In addition, the choice of whether or not to implement or select intentions not involved in mean-ends relationships can be made. For example, in Figure 24, the PC Product Provider can decide whether or not to Produce PC Products or Allow Peer-to-Peer Technology, even though these options are not specifically means to an end. In order to produce sensible evaluation results which take into account all aspects of the model, runs of the evaluation procedure typically involve placing initial values over alternatives for a goal (alternative selection for a goal) and placing values on intentions which are not alternatives of a goal. In order to distinguish between alternatives for a goal, alternative selections for a goal, alternative selections over multiple goals, and individual intentions within a model, we will call the latter an evaluation or analysis alternative. Each analysis alternative involves the selection of initial labels and then the application of the analysis procedure, including human judgment, in order to produce a set of analysis labels for all intentions connected to one or more of the initial analysis labels by (in this chapter) forward relations (links).

Definition: analysis alternative. *The results of a single run of the analysis procedure. Given a selection of initial analysis labels,* $v(i_1) \ldots v(i_n) \in \mathcal{V}$*, for some subset of intentions,* $i_1 \ldots i_n \in \mathcal{I}$*,* $n \leq |\mathcal{I}|$*, within an agent-goal model, the analysis algorithm produces analysis label results for a set of intentions,* $i_1 \ldots i_m \in \mathcal{I}$*,* $v(i_1) \ldots v(i_m) \in \mathcal{V}$*. The labeled intentions include those given initial labels; therefore the set of intentions with resulting labels is at least as large as the set of initial analysis labels,* $m \geq n$*, and in fact* $\{i_1 \ldots i_n\} \subseteq \{i_1 \ldots i_m\}$*. If a different set of initial analysis labels were used, this would produce a different analysis alternative, with potentially different label results over* $i_1 \ldots i_m$*.*

Typically, several analysis alternatives are applied to a single model, each exploring a different selection of initial values. This practice is described in more detail as part of the methodology for forward and backward analysis (Chapter 8).

In an i* model, any intention could be selected to receive an initial label as part of an analysis alternative (although leaf intentions are the most likely), and each initial value could be given one of six labels (although satisfied and denied are the most likely), the space of possible model analysis alternatives is large (if there are *n* intentions in the model, i.e. $|\mathcal{I}|$, there are 6^n possible sets of initial analysis labels over the model). However, evaluating an analysis alternative is not helpful unless it reflects some useful analysis question in the real world. In order to deal with the large space of choices in initial label selection, the starting points of a run of the analysis procedure should be determined by a relevant question being asked by the model. For example, in the Trusted Computing example in Figure 24, we can ask: "How does Obtaining PC Products from the Data Pirate affect Selling PC Products for Profit?" Several similar example analysis questions were provided in Sections 6.1 and 6.4.4. From this initial question we can assign satisfied to Obtain PC Products from the Data Pirate. This question, as with most analysis questions, leaves out information regarding some of the leaves in the model. For instance, it says nothing about Allowing Peer-to-peer Technology or Produce PC Products. However, in order to produce sensible analysis results, initial values for these intentions should be selected in line with the analysis question. In this case, Allow Peer-to-Peer Technology is allowed and PC Products are Produced. Also, the PC User does not Purchase PC Products, as it is acquiring them from the Data Pirate. This set of four initial labels forms the basis for one analysis alternative, initiating a single analysis of a model corresponding to our domain question. The initial values for the evaluation of this question are shown in Figure 27, while the results of the analysis alternative are shown in Figure 28. The remainder of this chapter provides details describing how the labels were propagated from Figure 27 to Figure 28.

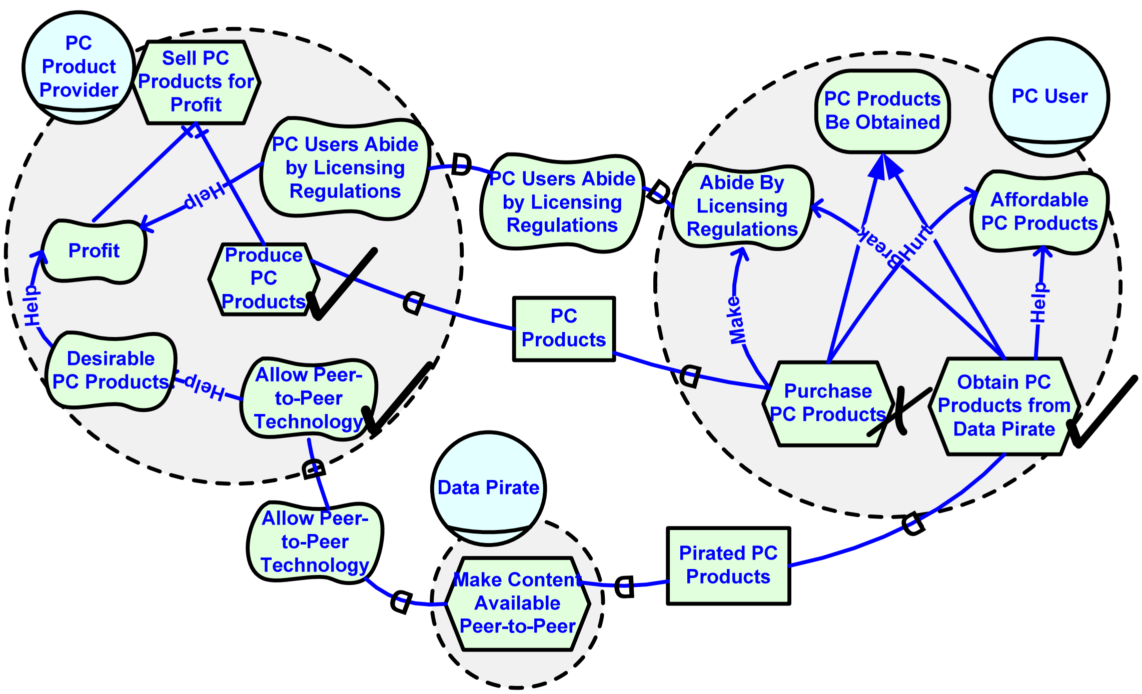


Figure 27: Example from the Trusted Computing Case Study in (Horkoff, 2006) showing Initial Analysis Labels

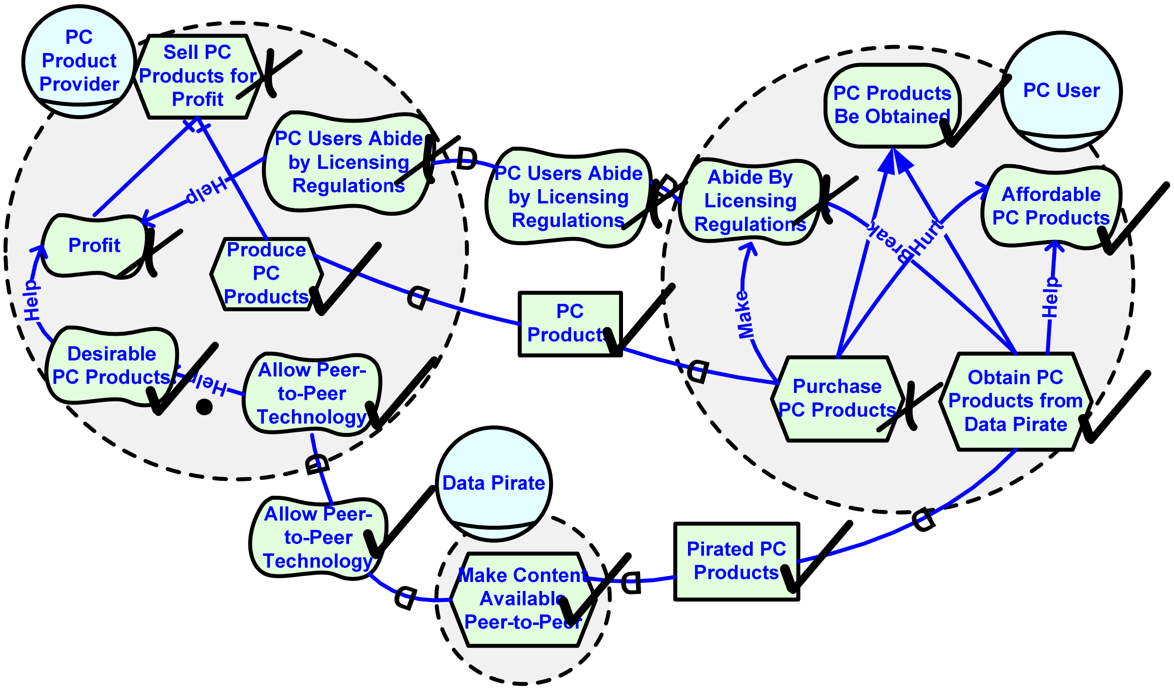


Figure 28: Example from the Trusted Computing Case Study in (Horkoff, 2006) showing Analysis Alternative Results based on Initial Labels from Figure 27

6.4.6 Evaluation Propagation Rules

We present rules in order to facilitate a standard propagation of values given a link type and contributing label in Step 2 of the procedure. These rules were originally defined in (Horkoff, 2006), but are redefined here using the formalism from Chapter 5. In order to express the forward propagation rules we develop axioms which express the results of each possible evaluation label when propagated through each type of relation in each direction. These axioms are similar to those defined in Tropos analysis, but are expanded to include constructs specific to i* (dependency links, conflict labels, unknown labels). Generally, for an intention $i \in \mathcal{I}$, with is the destination of a relationship, $r \in \mathcal{R}$, $r: i_1 \times \ldots \times i_n \rightarrow i$ these predicates take on the form:

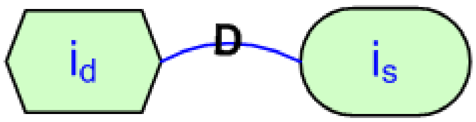
Forward Propagation:

(Some combination of $v(i_1) \ldots v(i_n),\ v \in \mathcal{V}) \rightarrow v(i)$

Dependency Links: The nature of a Dependency indicates that if the intention depended upon (*dependee*) is satisfied then the intention depended for (*dependum*) and intention depending on (*depender*) will be satisfied. Thus the analysis label of the dependee is propagated directly to the depender. We express propagation for these relationships in the axiom below. Recall that *s* is used to indicate the source of the relationship, while *d* indicates the destination.

Given: $r^{dep}: i_s \rightarrow i_d, v(i_s) \in \mathcal{V}$

$v(i_s) \rightarrow v(i_d)$



Decomposition Links: *Decomposition* links depict the intentions necessary to accomplish a task, indicating the use of an AND relationship, selecting the "minimum" label amongst the source labels. In order to facilitate this type of propagation, we must define minimum and maximum over our set of analysis labels, $\mathcal{V}$.

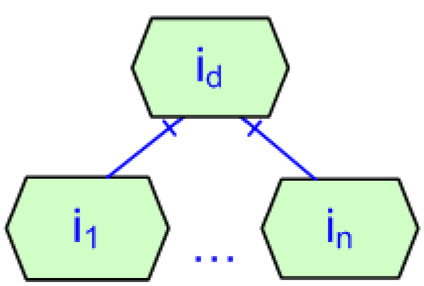
Definition: maximum label. *Given a set of analysis labels,* $v(i_1) \ldots v(i_n)$, $v \in \mathcal{V}$, *over* $i_1 \times \ldots \times i_n$, $i \in \mathcal{I}$, *the maximum label is the largest label, v, given the ordering in equation (2)* $(S(i) \geq PS(i) \geq U(i) \geq C(i) \geq PD(i) \geq D(i))$.

Definition: minimum label. *Given a set of analysis labels,* $v(i_1) \ldots v(i_n)$, $v \in \mathcal{V}$, *over* $i_1 \times \ldots \times i_n$, $i \in \mathcal{I}$, *the minimum label is the smallest label, v, given the ordering in equation (2)* $(S(i) \geq PS(i) \geq U(i) \geq C(i) \geq PD(i) \geq D(i))$.

From this we can define propagation over decomposition links:

Given: $r^{dec}: i_1 \times \ldots \times i_n \rightarrow i_d, v(i_1) \ldots v(i_n) \in \mathcal{V}$

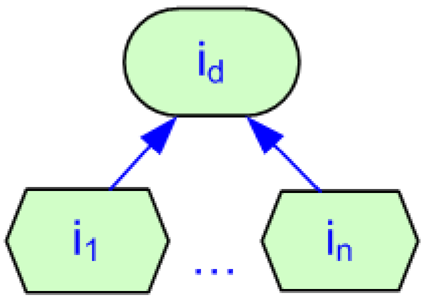
minimum$(v(i_1) \ldots v(i_n)) \rightarrow v(i_d)$



Means-Ends Links: Similarly, *Means-Ends* links depicts the alternative tasks which are able to satisfy a goal, indicating an OR relationship, taking the maximum values of intentions in the relation. To increase flexibility, the OR is interpreted to be inclusive.

Given: $r^{me}: i_1 \times \ldots \times i_n \rightarrow i_d,\ v(i_1) \ldots v(i_n) \in \mathcal{V}$

maximum$(v(i_1) \ldots v(i_n)) \rightarrow v(i_d)$



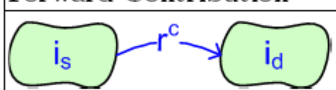
Contribution Links: We adopt the *Contribution* link propagation rules from the NFR procedure, as shown in Table 53. These rules intuitively reflect the semantics of contribution links. For instance, the Make link represents a positive contribution which is sufficient to satisfy a softgoal. Therefore this link propagates satisfied and partially satisfied labels as is. For negative evidence, links are treated as symmetric, in other words, if an intention Makes another intention when it is satisfied, it effectively Breaks this intention when it is denied. As a result, the Make link propagates denied and partially denied labels as is. Propagation rules for the Help link are similar, except that this link provides only a partial positive contribution. As a result, full evidence is weakened when passing through this link, although partial evidence remains partial (is not weakened to be non-existent). The propagation rules for the Break and Hurt links are roughly symmetric to Make and Help; positive evidence becomes negative and negative evidence becomes positive. In the case of a propagation of a denied label through a Break link (last row, Break column of Table 53), the result, as per the NFR Framework, is partially satisfied instead of satisfied, with the argument being that not achieving something negative produces only a partial positive result, not sufficient enough to fully satisfy an intention. We adopt this convention for consistency. The Some+ and Some- links are evaluated pessimistically, treating them as Help and Hurt links, respectively. Conflict and unknown labels always propagate without modification, unless through an unknown link, where a conflict becomes unknown. The absence of any value (None) is not propagated.

Table 53: 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	✗	✗.	✗.	✓.	✓.	✓.	?

The rules in Table 53 can be expressed using forward propagation axioms, similar to the axioms described for dependency, decomposition, and means ends links. Generally, given the type of contribution link, $r^c \longmapsto \{R^m, R^{hlp}, R^u, R^{hrt}, R^b\}$, and the source label, $v(i_s)$, a rule for each row/column combination of Table 53 of the form $v(i_s) \rightarrow v(i_d)$, can be defined. In fact, we can use the partial orders from equation (1) to simplify these rules from 42 (6 rows * 7 columns) to 14. The simplified rules are listed in the table below. We list these rules by $V(i_s)$, the value of the source intention. Results are given for each $r \in R^c \longmapsto \{R^m, R^{hlp}, R^u, R^{hrt}, R^b\}$.

Table 54: Forward Propagation Axioms for Contribution Links

Forward Contribution	
	
$\mathbf{V(i_s)}$	$\mathbf{V(i_s) \rightarrow V(i_d)}$
U	$c = any : U(i_s) \rightarrow U(i_d)$
C	$c = any : C(i_s) \rightarrow C(i_d)$

$\mathbf{V(i_s)}$	$\mathbf{V(i_s) \rightarrow V(i_d)}$	
S	$c = m : S(i_s) \rightarrow S(i_d)$	$c = b : S(i_s) \rightarrow D(i_d)$
	$c = hlp : S(i_s) \rightarrow PS(i_d)$	$c = hrt : S(i_s) \rightarrow PD(i_d)$
PS	$c = m, hlp : PS(i_s) \rightarrow PS(i_d)$	$c = b, hrt : PS(i_s) \rightarrow PD(i_d)$
PD	$c = m, hlp : PD(i_s) \rightarrow PD(i_d)$	$c = b, hrt : PD(i_s) \rightarrow PS(i_d)$
D	$c = m : D(i_s) \rightarrow D(i_d)$	$c = b, hrt : D(i_s) \rightarrow PS(i_d)$
	$c = hlp : D(i_s) \rightarrow PD(i_d)$	
$v \in V$	$c = u : v(i_s) \rightarrow U(i_d)$	

6.4.7 Resolving Multiple Contributions

Softgoals are often the recipient of multiple contribution links. We adopt the notion of a "Label Bag" from (Chung et al., 2000), used to store all incoming labels for a softgoal. Labels in the label bag are resolved into a single label in Step 3, either by identifying cases where the label can be determined without judgment, or by human judgment. The former cases are described in Table 55. If the bag has only one label (case 1) the results is that label. If the bag has multiple full labels of the same polarity (case 2) or multiple labels of the same polarity with one full label (case 3), the result is the full label. If the human judgment matching the label bag has already occurred, the previous answer will be used (case 4). Finally, if a previous human judgment produced a full label, and the label bag has become more positive or more negative matching the polarity of the full label, the result is automatically the same full label (case 5). For example, in Figure 23, the Immediacy [Service] softgoal in Kids and Youth receives a satisfied and a partially satisfied label from incoming contributions links, resolved to a satisfied label using Case 3 in Table 55, reflecting the idea that evidence propagated to softgoals is roughly cumulative.

Table 55: Cases where Overall Softgoal Labels can be Automatically Determined

Label Bag Contents	Resulting Label
1. The bag has only one label. Ex: {✗.} or {✓.}	the label: ✗. or ✓.
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 human judgment situation has already occurred for this element and the answer is known	the known answer
5. A previous human judgment situation for this element produced ✓ or ✗, and the new contribution is of the same polarity	the full label: ✓or ✗

6.4.8 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 55. We can formally define what it means for an intention to require human judgment. **Definition: need for human judgment.** An intention, $i \in \mathcal{I}$, needs human judgment if:

- i is the recipient of more than one incoming contribution link, i.e. there exists an r_1 and r_2 $\in \mathcal{R}$ such that $r^c_1 : i_1 \rightarrow i$ and $r^c_2 : i_2 \rightarrow i$, AND:
  - There is a conflict between labels, as defined in Section 6.4.3.1.
  - Or, $PS(i)$ or $PD(i)$ holds and i has not received a human judgment in the current algorithm iteration

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 intentions should be used.

For example, the resulting label for Happiness [Counselors] in Figure 23 is determined by human judgment. This softgoal receives partially denied labels from Avoid Burnout and High Quality Counseling, but receives a partially satisfied label from Help as many Kids as Possible, according to the propagation rules in Table 53. Here, using our knowledge of the domain, we decide that

Counselors would be mostly unhappy, labeling the softgoal as partially denied. Situations such as this would be good areas for potential discussions with stakeholders involved in the modeling process.

When recording a human judgment, the judgment can be stored as a new propagation axiom reflecting the decision of the user(s). In the example above, the following axiom would be added:

$$PD(\text{Avoid Burnout}) \wedge PD(\text{High Quality Counseling}) \wedge PS(\text{Help as many Kids as Possible}) \rightarrow PD(\text{Happiness [Counselors]})$$

The utility of interactive judgments is tested with various empirical studies in Chapter 12.

6.4.9 Combinations of Links

Intentions in i* are often the destination of more than one type of link. This occurs when an intention is the recipient of a *Dependency* link and a *Means-ends*/*Decomposition* link or a *Contribution* link. “Hard” links (Decomposition, *Means-Ends*, and *Dependency*) are combined using an AND of the final results of each link type. This can be described formally similar to the decomposition rules in Section 6.4.6. 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. An example of this type can be seen in High Quality Counseling in the Organization.

6.4.10 Incomplete Labels

In the forward analysis 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 evaluation labels for an intention may change throughout the procedure and the same softgoal may require human judgment multiple times. Section 4.2.2.1 in (Horkoff, 2006) discusses implementation choices concerning resolving labels with only partial incoming information. Alternative implementations include designing the algorithm to wait until all incoming labels arrive, or to resolve and propagate only labels present during each step. Both the approach in (Horkoff, 2006) and the forward algorithm in this work take the second approach. The first approach greatly complicates the algorithm, and runs the risk of deadlock while waiting for incoming labels which may depend on the label currently being considered. Implementation in the current work goes beyond the work of (Horkoff, 2006) by storing each previous judgment, applying judgments automatically if they have already occurred. More details concerning this and other implementation features can be found in Chapter 11.

6.4.11 The Evaluation Algorithm

The algorithm adopts the structure outlined in the NFR procedure by including iteration over two steps: propagation and value resolution. In the first step, all present labels are propagated through all outgoing links using the rules described in the previous section. In the second step, the resulting evaluation values for softgoals are determined, using either the automatic cases in Table 55, or human judgment.

Once the values for all intentions have been determined in the second step of the algorithm, the cycle starts again. The labels to be propagated are kept track of using a queue of intentions to which the labels are assigned, $\mathcal{LQ}$, starting with the initial labels, and adding each final label produced in step 1 and 2. The algorithm will terminate when all labels have been propagated and this queue is empty.

As the procedure allows the placement of initial values, $v(i_1) \ldots v(i_n) \in \mathcal{V}$, on non-leaf nodes, it is necessary to define how these values are affected by subsequent propagation. In the case of hard intentions (non-softgoals), subsequent propagation overrides the initial value. In the case of softgoals, initial values are placed in the bag of labels, leaving conflicts between initial and propagated values to human judgment. The implementation retains initial labels for each intention.

Pseudocode describing the evaluation algorithm is shown in Table 56. As our eventual implementation is object-oriented, we use a system of objects and attributes to describe the intentions, relations and analysis values in the pseudocode. For example, instead of using the *v(i)* notation to indicate the analysis label for an intention, *i*, we use `i.v`, indicating that the label is stored as an attribute of an intention. We refer to the initial label for each intention as an attribute `i.il`, $i \in \mathcal{I}$. The type of each intention in the set intention type is referenced by an attribute, `i.type`. The label bag for each intention is an attribute, `i.LB`. Each label bag has a Boolean attribute, `i.LB.r`, indicating whether or not it has been resolved since the last time its contents have changed. The algorithm stores a list of all the human judgments made in the `HJ` list.

The algorithm starts with the set of all intentions, `I`, relations, `R`, and the set of initial labels, `IL`. It iterates over steps 1 and 2 as described, until the label queue is empty. In step one, each label to propagate is removed from the label queue and the resulting propagated value is calculated (findResultingEvalValue). The algorithm uses methods `ContributionRules`, `Means-Ends Rule`, and `Decomposition Rule`, referring to the propagation rules described in Section 6.4.6. If the label to propagate is for a softgoal, the resulting value is added to the label bag for that intention. Otherwise the value is added directly to the model and the label queue.

In step 2, each unresolved label bag is resolved, either using automatic cases or human judgment (PromptUser). The results are added to the label queue. Additional helper methods which initialize the label queue, apply propagation, or manage the label bag are also included.

The procedure for resolving a mix of hard links could be simplified using attributes and data structures for hard intentions. Here, we chose a repetitive method of examining all hard links for each incoming value; we leave this to be optimized in the implementation.

Table 56 Pseudocode of the i* Evaluation Algorithm

```
qualitativeInteractiveEvaluation(I, R, IL):
init(LQ, IL);
While !LQ.empty()
  step1(LQ)
  step2(LQ)
init(LQ, IL):
  For each i.v ∈ IL
    i.il = i.v; LQ.push(i)
    If i.type == Softgoal
      i.LB.addToBag(i.v, "Initial")
step1(LQ):
  LQ2 = LQ;
  While !LQ2.empty()
    i_s = LQ2.pop()
    LQ.pop()
    For each r ∈R s.t. r:i_s→i_d
      Label v = findResultingEvalValue(r, i_s, i_d)
      If i_d.type = Softgoal
        i_d.LB.addToBag(v, i_s)
```

```
      Else if i_d.il == N and v != N
        i_d.v = v
      If i_d ∉LQ
        LQ.push(i_d)
step2(LQ):
  For each i ∈ I s.t. i.type == Softgoal and i.LB.r == false
    i.v =AutomaticCases(e.LB)
    If (i.v == N)
      If <i, i.LB, v> ∈HJ
        i.v = v
      Else if <i, i.LB, *> ∉HJ
        i.v = PromptUser(i.LB)
        HJ.add(<i, i.LB, i.v>)
    If i ∉LQ
      LQ.push(i)
    i.LB.r = true
findResultEvalValue(l, i_s, i_d):
  If r ⟼ r^c
    Label v = ContributionRules(r, i_s)
  If i_d.type == softgoal
    Label v = i_s.v
  Else
    Label v = resolveMixofHardLinks(i_d)
  return v
resolveMixofHardLinks(i_d):
  Label v
  For each r ∈R s.t. l: (i_1,…,i_n)→i_d and r not ⟼ r^c
    v = min(v, resolveSingleHardLinks(r))
  return v
resolveSingleHardLinks(l):
  If r ⟼ r^me
    Label v = Means-Ends Rule(i_1.v,…,i_n.v) where r^me: (i_1,…,i_n)→i_d
  If r ⟼ r^dec
    Label v = Decomposition Rule(i_1.v,…,i_n.v) where r^dec: (i_1,…,i_n)→i_d
  If r ⟼ r^dep
    Label v = i_s.v
  return v
addToBag(v, i):
  i.LB.remove(<*, i>)
  i.LB.add(<i.v, i>)
  i.LB.r = false
```

6.4.12 Model Cycles, Termination, and Running Time

Goal models often contain cycles, values which indirectly contribute to themselves. See the relationships between Happiness [Counselors] and High Quality Counseling in Figure 23 for an example cycle involving help links. Often these situations will converge to particular value, but in some situations they may fluctuate between values indefinitely.

We need to ensure that values do not continually fluctuate, causing an infinite loop. Take for example the model snipped in Figure 29. In evaluating a loop of the model such as this, evaluation always begins with an initial or contributed label, for example the satisfied *(S)* label for A, (*v(A)* = *S*), at the top of Figure 29. This label is stored and presented to the user in each human judgment situation, the user may pick any value they deem appropriate. If an intention receives a contribution from an intention for which it already has a contribution, the old contribution is replaced by the new. As human judgment situations are stored, if a bag state requiring human judgment has arisen previously, the user is not prompted again. Given these aspects of the algorithm, an infinite loop may occur. Take the example shown in Figure 29 with a trace of the algorithm including propagation rules, judgments, and the label queue in Table 57. When representing the label queue, `LQ`, we show both the intention whose label is due to be propagated and the label itself, in the form `<i, i.v>`. In this figure, the new label in each step is highlighted in red. In this example, after the 5th iteration, the trace loops back to the same state as the 2nd iteration, looping infinitely between the 2nd to 5th iteration.

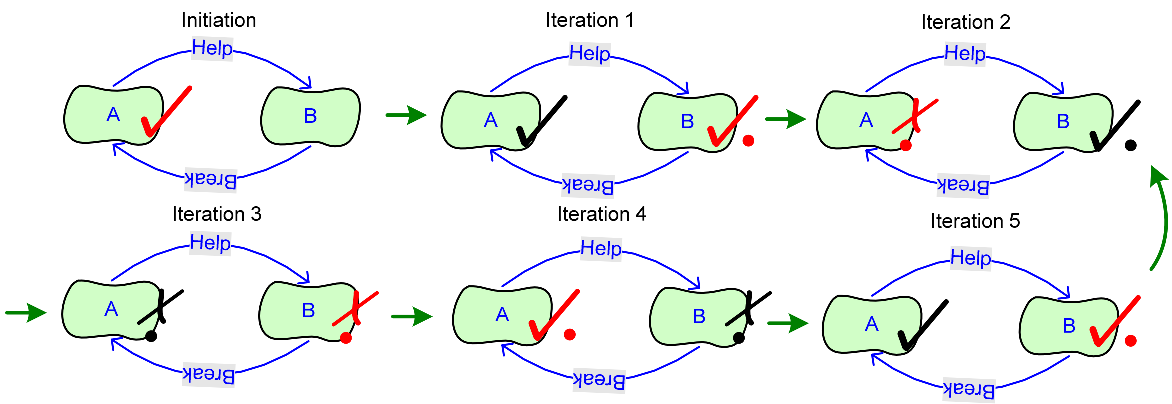


Figure 29 Example of Infinite Loop in Algorithm

Table 57 Trace of Infinite Loop in Algorithm

Iteration	Rule	Human Judgment	Resulting LQ
Init			{<A, S>}
1	Contribution Rules(S, Help) = PS		{<B, PS>}
2	Contribution Rules(PS, Break) = PD	Prompt User(A.LB = {<S, N>, <PD, B>) = PD	{<A, PD>}
3	Contribution Rules(PD, Help) = PD		{<B, PD>}
4	Contribution Rules(PD, Break) = PS	Prompt User(A.LB = {<S, N>, <PS, B>}) = PS	{<A, PS>}
5	Contribution Rules(PS, Help) = PS		{<B, PS>}
6 (same as 2)	Contribution Rules(PS, Break) = PD	Human Judgment Situation (A.LB = {<S, N>, <PD, B>}) has Occurred = PD	{<A, PD>}
Loop from 2 to 5			

To avoid such situations, we implement the relatively shallow solution of storing a count of each of the combinations of source intentions that have been placed in the label queue along with their current labels, for example <i1, i1.v = PS> may have occurred three times. Once this count has reached a certain fixed number, *r*, the same source and label combination cannot be placed in the label queue again. This solution allows for a certain number of value fluctuations for non-looping situations, such as in Figure 30, but will put a cap on the number of iterations which can occur in situations such as in Figure 29. In this way, if there are *n* intentions in the model, supporting a total of *6* analysis labels and a cap of *r* times in the label queue, the label queue has a maximum lifetime size of *6rn*, and the algorithm must terminate. The running time of the algorithm is *O(n)*, where *n* is the size of the model.

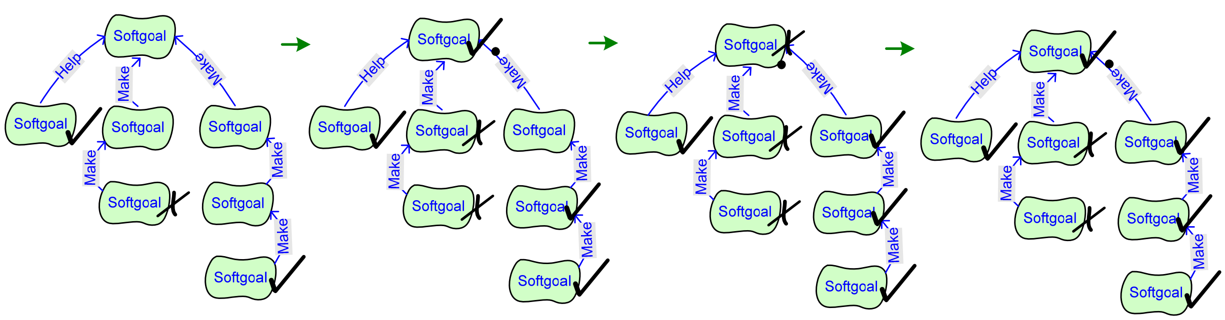


Figure 30 Non-Looping Value Fluctuation Example, allowed if r>2

Section 4.4.1 in (Horkoff, 2006) contains an additional example of a model cycle and label fluctuation.

Experience has shown that during manual application of the procedure the presence of fluctuating cycles becomes apparent to the evaluator after a few iterations, allowing the evaluator to select an appropriate converging value or stop the analysis.

6.5 Analysis Examples

In order to illustrate the analysis procedures described in this chapter, we provide two analysis examples. A third example can be found in Section 4.5 in (Horkoff, 2006). The first example is described in prose, for simple comprehension. The second example contains a detailed trace of the label bag and can be found in Appendix C if more detail is required.

Counseling Service Example. In the first example, we evaluate one of the counseling options in Figure 23, asking "What is the effect of using a Cybercafe/Portal/Chat Room?" Results shown in Figure 31, below, can be analyzed from the point of view of each actor. For Kids and Youth, the Cybercafe/Portal/Chat Room provides Immediacy as well as a Comfortable Service, but jeopardizes Anonymity, making the overall assessment weakly satisfied for Get Effective Help. From the point of view of Counsellors, the alternative has a positive effect on Help as Many Kids as Possible, but has a negative effect on Burnout and the Quality of Counselling. From the point of view of the Organization, the service also has a positive effect on Helping as Many Kids as Possible and Immediacy, but has a negative effect on Anonymity, Avoiding Scandal, Increasing Funds, and the Quality of Counselling. There is conflicting evidence for the ability to Help Kids. Overall, this alternative is judged to be not viable. A further round of evaluation is needed to assess the other alternative in the model, text messaging, and to use the goals in the model to brainstorm further online counselling services which balance concerns more effectively.

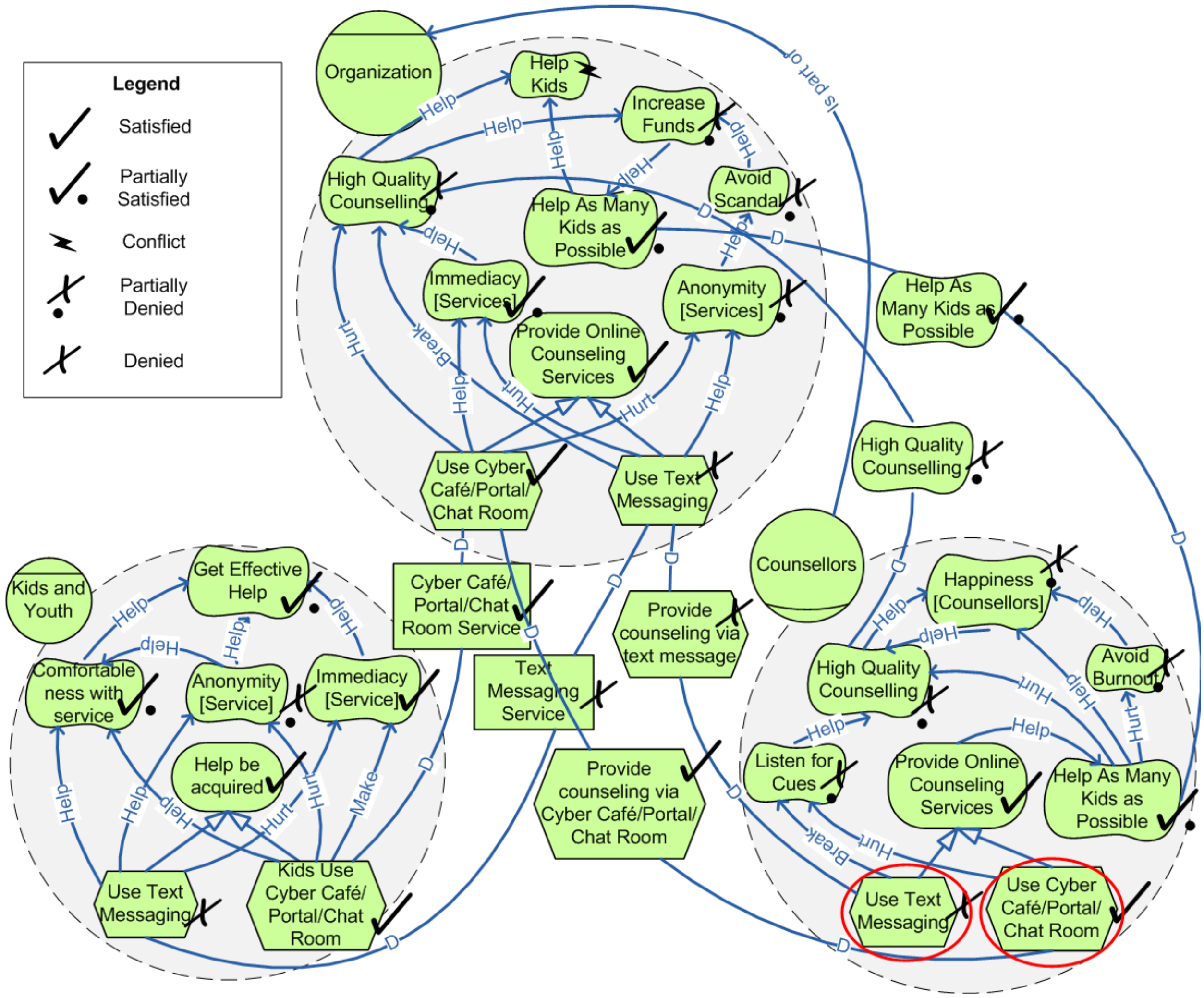


Figure 31 SR Model for Youth Counselling showing Final Evaluation Results

6.6 Conclusions

We have previously considered the contributions made by the interactive forward procedure introduced by Horkoff (2006). In this chapter we have revisited this procedure, providing a formal definition for procedure concepts, propagation and human judgments. We have revised our consideration of and mechanisms for procedure termination. One of the primary contributions of this chapter is to describe previous work forming a basis for the current framework more precisely. We also describe how to represent and store human judgments. We summarize the contributions of this chapter to the contributions of Horkoff (2006) in Figure 32. We increase the summary value of the R7 Provide Definition requirement, and the R10 Capture Human Judgments requirement.

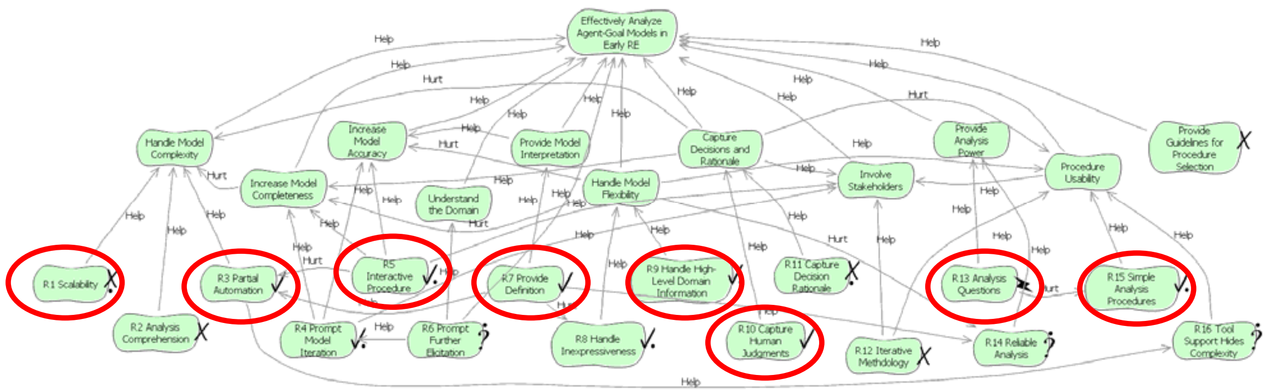


Figure 32 Satisfaction Analysis for the Requirements for Analysis of Agent-Goal Models in Early RE based on the Contributions of Horkoff (2006) and Chapter 6

Chapter 7 Iterative, Interactive, Backward Satisfaction Analysis of Agent-Goal Models

The previous chapter has introduced an interactive analysis procedure propagating forward from alternatives allowing users to ask "What if?" questions. In this chapter we introduce a backwards, iterative, interactive analysis procedure propagating backward from high-level target goals, allowing users to ask "Is this possible?" questions. The approach is novel in that it continues the axiomatization of propagation in the i* framework, including the role of human intervention to potentially resolve conflicting contributions or promote multiple sources of weak evidence. A simple application example is used to illustrate the procedure. Application to larger-scale examples are described in Chapter 12. We end with a discussion of our decisions in implementing the procedure including ways in which the algorithm could be expanded.

The procedure introduced in this chapter contributes to several of our requirements for early RE analysis of agent-goal models listed in Chapter 3. Most notably, we aim to expand the analysis questions (R13) supported by analysis. In addition, we produce a partially automated procedure in order for analysis to operate over complex models (R3). As in the forward procedure, the user is prompted for judgment over conflicting or partial areas of the model (R10), making the procedure interactive (R5). Procedure concepts including propagation and judgment are defined formally (R7); however our formalism allows for the representation of high-level domain concepts (R9). We argue that the procedure is scalable (R1) over models of a reasonable size,

This chapter is an expansion of the following papers/reports:

Horkoff, J. (2008b). CSC2108 Course Project: Qualitative, Interactive, Backward Analysis of i Models.*

Horkoff, J., & Yu, E. (2008). Qualitative, Interactive, Backward Analysis of i Models. 3rd International i* Workshop (pp. 4-46). CEUR-WS.org.*

Horkoff, J., & Yu, E. (2010a). Finding Solutions in Goal Models: An Interactive Backward Reasoning Approach. Conceptual Modeling (ER 2010), 29th International Conference on Conceptual Modeling, Vol. 6412, p. 59. Springer-Verlag New York Inc.

with further running time tests provided in Chapter 11. The procedure is designed to be relatively simple; the details of the formal representation are hidden from the user (R15). The comprehensibility of the analysis results (R2) will be addressed in Chapter 9. The simplicity of the procedure implementation (R15) and the ability of the procedure to prompt model iteration (R4) will be explored in Chapter 12. We highlight the intended contributions of this Chapter in Figure 33.

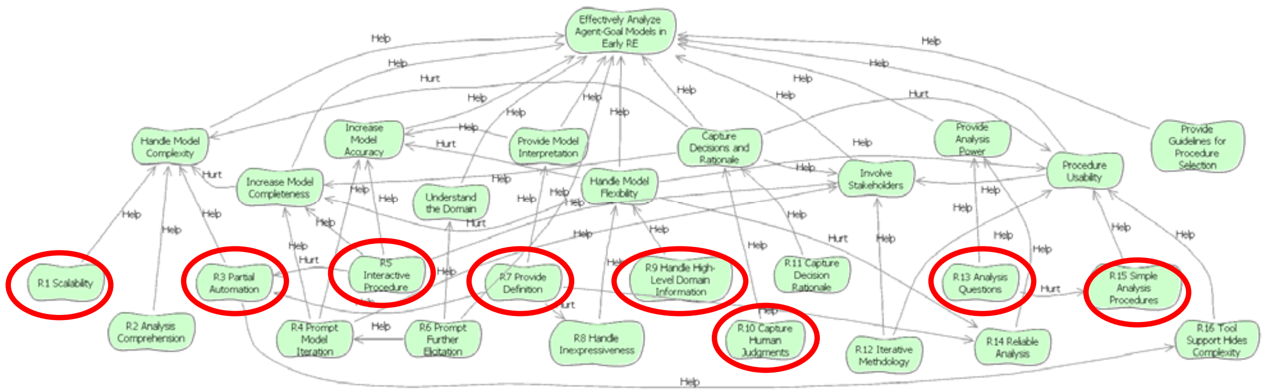


Figure 33 Focus of Chapter 7 Concerning the Requirements for analysis of Agent-Goal Models in Early RE

7.1 The Need for Backward Interactive Analysis

In addition to "What if?'" questions, users also want to be able to answer questions such as "Is this goal achievable?", "If so, how?", and "If not, why?" To illustrate forward and backward analysis we use a simple model of a generic application shown in Figure 34. We use an example which is simpler than the counseling service and trusted computing examples used previously, due to the underlying complexity of the backward procedure implementation. In the figure, the application needs to Implement Password System with two options identified: Restrict Structure of password, for example, must have a numeral and be more than five characters, and Ask for Secret Question, in case of password loss. These options are not mutually exclusive. The overall goal is to Attract Users, which is helped by both Security and Usability. Restrict Structure of Password makes Security (according to this model), but hurts Usability, while Ask for Secret Question helps Usability.

In analyzing this model, users would want to ask "what if?" forward-type questions, analyzing the effectiveness of each feasible alternative. In this case there are three feasible analysis alternatives, with one or the other alternative selected, or both. For example, in Figure 34 we evaluate the alternative where Ask for Secret Question is satisfied but Restrict Structure of Password is denied. The first pass of the procedure propagates values to Implement Password System and Security automatically, but prompts the user for human judgment on Usability, with incoming values of partially satisfied from Ask for Secret Question and partially satisfied from Restrict Structure of Password. In this case the user decides Usability is partially satisfied. Next they are asked about Attract Users, receiving values of partially denied and partially satisfied. The user decides that with only partial Usability and no Security, Attract Users is partially denied.

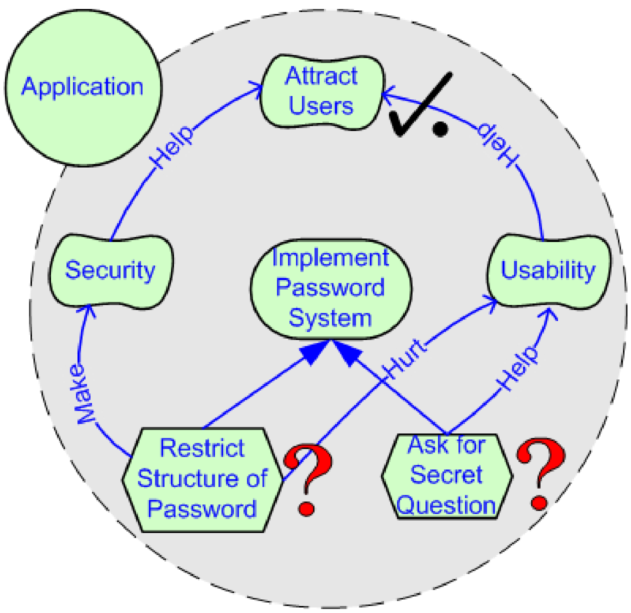


Figure 34 Example i* model for a Generic Application

Although individually evaluating the effectiveness of each analysis alternatives is useful, we would like to be able to ask other forms of questions. Specifically, instead of being driven by bottom-up analysis, evaluating each alternative, we would like to be able to ask "Is this possible?" questions, where constraints over the model are provided and a viable alternative is provided. Over our example model, we would like to ask questions such as: is it possible for Attract Users to be at least partially satisfied, and if so, how? To answer this type of question, we need a "backward" procedure which starts at the intention(s) of interest and, using the same propagation rules as the forward procedure when possible, works down the links in the model, looking for potential solutions. In order to satisfy the interactive requirements of early RE analysis, this procedure should prompt for human judgment in situations where labels cannot be determined without human input, making the operation consistent with the forward procedure.

In addition, if it is not possible to obtain constraints placed over the model, it would be useful to know “why?", identifying the areas of the graph involved preventing the constraints from being met.

One way to find answers for “is this possible?” questions would be to apply the forwards procedure repeatedly and exhaustively for all reasonable alternatives until either the desired values are produced, or not. However, this approach could be tedious and laborious, especially for larger models with many alternatives. We can also consider finding alternatives which match model constraints manually, similar to our consideration of manual forward analysis in Section 6.1. This would require tracing links and assigning potential labels backwards, manually. In each backward step involving multiple incoming links multiple sets of labels can be possible. Users would have to select a set, work down to assess resulting values in terms of leaf intentions, and then backtrack if these values were not satisfactory. This is a cognitively difficult task, especially for large models, and is in fact far more difficult than manual forward analysis. A procedure automating backward propagation and backtracking through values is needed. In the rest of this chapter we describe a procedure providing this type of analysis questions for agent-goal models.

7.2 Challenges in Backward Analysis

The procedure introduced in this chapter encodes forward and backward propagation rules in conjunctive normal form (CNF), iteratively applying a SAT solver and human intervention to search for an acceptable solution. In formulating such an interactive backward procedure we face some interesting questions and technical challenges. What types of questions could and should be posed to the user, and at what point in the procedure? How can the encoding be modified to reflect human judgment? What is added? What is removed? When a choice does not lead to an acceptable solution, to what state does the procedure backtrack? As information is lost in forward propagation when evidence is manually combined, what assumptions about this evidence can be made when propagating backward? How can the axiomization deal with explicit values of conflict and unknown, compared to approaches that only allow for positive and negative values (Giorgini et al., 2004b)? Can we find a balance between constraining the problem sufficiently to avoid nonsensical values and allowing enough freedom to detect the need for human judgment? How can we use information about SAT failures to inform the user? Is there a computationally realistic approach? The procedure introduced in this chapter represents one approach to answering these questions.

7.3 Background: SAT and Unsatisfiable Core

SAT solvers are algorithms which accept a Boolean formula in conjunctive normal form (CNF), composed of a conjunction of clauses. The algorithm searches for a truth assignment of the formula's clauses to make the formula true. It does so by making a series of decisions concerning the values of variables, backtracking if a decision proves to be not viable. If a solver can find a satisfying assignment, it returns only one such assignment, saying nothing about the presence of other permissible answers. An example logical formula converted to CNF is shown below. SAT problems are represented in a dimacs file, where variables are converted to positive and negative numbers. A short example is shown for the given problem. The SAT solver finds a solution for this formulation as shown.

Original clause:

$p \rightarrow (q \wedge s)$

CNF Conversion:

$(\neg p \vee q) \wedge (\neg p \vee s)$

Dimacs Representation

-1 2 0

-1 3 0

SAT Answer:

-1 2 3

$\neg p \wedge q \wedge s$

Although the SAT problem is NP-Complete, algorithms and tools that can solve many SAT problems in a reasonable amount of time have been developed, for example, the zChaff tool (Mahajan, Fu, & Malik, 2004), used in this work.

When a SAT solver fails to find a satisfying assignment, it is useful to know about the underlying conflict(s). Further improvements on SAT algorithms have resulted in the ability to find an unsatisfiable core, a list of clauses in the CNF which result in a conflict. These clauses can be used to form a resolution proof, showing how the clauses work together to produce a conflict, i.e., *(a* $\wedge \neg$*a)*. Finding a minimal unsat core is a computationally difficult problem (Zhang, Li, & Shen, 2006), but many approaches exist for finding a small but not minimum core (for example (Bruni & Sassano, 2001)). We use the zMinimal application provided with zChaff to find a small but not minimal unsat core, showing the user a list of intentions included in the set of conflicting clauses when the SAT solver fails to find a solution. More information about visualization techniques used to highlight the unsatisfiable core intentions in a model can be found in Chapter 9.

7.4 Qualitative, Interactive, Iterative, Backward Analysis for Agent-Goal Models

In this section, we provide an overview of the backward analysis algorithm. Restrictions over agent-goal models required in the analysis are described using our previous formalism. We describe how to express an i* model and constraints in a form that can be used with a SAT solver. Axioms expressing backward propagation over i* models are provided. The interactive nature of the procedure is described by outlining the iterative use of human judgment over analysis results. We provide pseudocode for the backward algorithm. A detailed example is given over the example model used in this chapter. Finally, we describe running time, termination guarantees, soundness and completeness for the procedure.

7.4.1 Procedure Overview

The approach encodes the model in a SAT formula, and then iteratively runs the SAT solver, prompting the user for input regarding intentions which required human judgment after each run. When human judgment is no longer needed and a satisfying assignment is found, the procedure ends, providing an answer. If a satisfying assignment is not found the procedure tries to backtrack over human judgments. If a satisfying assignment is not found and no further human input can be given, the procedure ends, informing the user that the target is not possible. The choice of SAT as an implementation tool is discussed in Section 7.5.

The procedure has been implemented in the OpenOME Tool ("OpenOME, an open-source requirements engineering tool," 2010). Characterizing agent-goal model propagation in CNF requires a more formal definition for agent-goal model (in our case, i*) concepts, including evaluation values (intention labels). We have addressed the need for this formalism in Section 5.4. In Chapter 6 we have provided axioms expressing i* propagation in the forward directions. In the next sections, we provide similar axioms for the backward direction, including necessary constraints over evaluation values, and the encoding of human judgment. We describe the use of a SAT solver in an iterative procedure in more detail, using our simple example to illustrate.

7.4.2 Restrictions on Agent-Goal Model

In order to produce an agent-goal model which can be more easily translated into CNF form and to ensure the convergence and termination of the algorithm, we place the following restrictions on an i* model:

- *Each intention has at most one Decomposition, Dependency or Means-Ends relation which determines its level of satisfaction or denial, i.e.,* $\forall i \in \mathcal{I}$, *only one of* $R^{dep}: \mathcal{I} \rightarrow i$, $R^{dec}: \mathcal{I} \times \ldots \times \mathcal{I} \rightarrow i$, *or* $R^{me}: \mathcal{I} \times \ldots \times \mathcal{I} \rightarrow i$ *holds for i.*

- *The model must have no cycles, i.e., for every path in the model,* $r_1, \ldots, r_n \in R$, r_1: $i_1(\times \ldots \times \mathcal{I}) \rightarrow i_2$, r_2: $i_2(\times \ldots \times \mathcal{I}) \rightarrow i_3, \ldots, r_{n-1}$: $i_{n-1} (\times \ldots \times \mathcal{I}) \rightarrow i_n$, i_k *must not equal* i_j, *for* $1 < i, j < n$.

7.4.3 Expressing Qualitative, Interactive Propagation in CNF

To express the problem of assigning evaluation labels to an agent-goal model in terms of a CNF SAT formula, we follow the formalization in (Giorgini et al., 2004b), adopting their classification of the components of the formula as follows:

- The target values for the procedure, *ΦTarget*
- Axioms describing forward propagation, *ΦForward*
- Axioms describing backward propagation, *ΦBackward*
- Axioms describing invariant properties of evaluation labels, *ΦInvariant*
- Any additional constraints on propagation, *ΦConstraints*

The SAT formula is constructed as follows:

$\Phi = \Phi Target \wedge \Phi Forward \wedge \Phi Backward \wedge \Phi Invariant \wedge \Phi Constraints$

Target. The target for an evaluation is simply a conjunction of the desired values for each target intention. In our example question over our example model in Figure 38 the target would be *PS(Attract Users)*. We could constrain the target further by saying that the target should only have that value, for example if our target is *PS(i)*, we add ¬*C(i)* and ¬*U(i)* and ¬*PD(i)*, but we want to allow for targets to have conflicting values, making them candidates for human intervention.

Invariant. As invariant axioms, we include the partial order defined in Section 6.4.3.2, repeated below.

$$\forall i \in I: \quad S(i) \geq PS(i) \Leftrightarrow S(i) \rightarrow PS(i) \qquad (1)$$
$$D(i) \geq PD(i) \Leftrightarrow D(i) \rightarrow PD(i).$$

Constraints. When using the analysis procedure, the user could add any additional constraints into the SAT formula, following the approach of (Giorgini et al., 2004b). In our example, we constrain leaf intentions such that these intentions must be assigned one of the six evaluation labels, as below:

$\forall i \in \mathcal{I}$, *s.t. i is a leaf:* $PS(i) \vee C(i) \vee U(i) \vee PD(i)$

Restricting the model formalization in this way ensures that the answer provided by the SAT solver applies labels to all connected intentions. In our example, we would add these constraints for our two leaf intentions, Restrict Structure of Password and Ask for Secret Question.

7.4.4 Backward Propagation Axioms

In order to express the forward and backward propagation rules we develop axioms which express the results of each possible evaluation label when propagated through each type of relation in each direction. The forward axioms have been described in Chapter 7. We review their form along with the introduction of the backward axioms, below. Generally, for an intention $i \in \mathcal{I}$, $R: i_1 \times \ldots \times i_n \rightarrow i$ these predicates take on the form:

Forward Propagation:

(Some combination of $v(i_1) \ldots v(i_n)$*,* $v \in \mathcal{V}) \rightarrow v(i)$

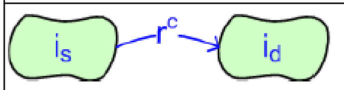
Backward Propagation:

$v(i) \rightarrow$ *(Some combination of* $v(i_1) \ldots v(i_n)$*,* $v \in \mathcal{V})$

The backward propagation axioms can be derived from examining the propagation rules described in Section 6.4.6. For dependency, decomposition, and means-ends links, the backward propagation rules are identical to the forward, but in the opposite direction. For example, in a means-ends relationships with two sources *b* and *c* to destination *a*, either *b* or *c* must be satisfied for *a* to be satisfied in the forward direction, $(S(b) \vee S(c)) \rightarrow S(a)$. In the backward direction, if *a* is satisfied, then either *b* or *c* must be satisfied, $S(a) \rightarrow (S(b) \vee S(c))$. The SAT solver will try to find a satisfying assignment where either $S(b)$ or $S(c)$ or both are true. The general form for forward and backward propagation of full satisfaction for means-ends links with *n* sources and destination i_j is $(\bigvee_{j=1}^{n} S(i_j)) \rightarrow S(i_d)$ and $S(i_d) \rightarrow (\bigvee_{j=1}^{n} S(i_j))$, respectively. The other axioms for means-ends or decomposition use similar reasoning. We list only the forward axioms for these links in Table 58. Axioms in the table have been simplified when possible using the invariant clauses in Equation (1).

Propagation axioms for contribution links are treated differently. In the forward direction when an intention, *i*, is the recipient of multiple contribution links (there exists an $r_1 \ldots r_n \in \mathcal{R}$ such that $r^c_1: i_1 \rightarrow i \ldots r^c_n: i_n \rightarrow i$), each link from source to destination, r_j for *j* from $1 \ldots n$, contributes a label. As described in Section 6.4.8, these labels are combined into a single label using either automatic rules or human judgment. In the backward direction a single destination label for *i*, $v(i_d)$ is used to place constraints on the values of one or more sources, $v_j(i_j) \in \mathcal{V}$, for *j* from $1 \ldots n$. We can only make minimal reasonable assumptions concerning the labels of the source intentions given the label of the destination intention. For example, if $v(i_d) \longmapsto PS$, we assume that at least one of the incoming values is PS, meaning that one of the positive links propagates at least a PS value (i.e. $\exists j,\ r_j \in Pos,\ s.t.\ v_j(i_j) \longmapsto PS$) or one of the negative links propagates at least a PD value (i.e. $\exists k,\ r_k \in Neg,\ s.t.\ v_k(i_k) \longmapsto PD$). The rest of the backward assumptions are similar.

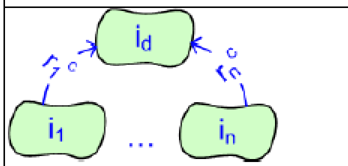
Forward Contribution



$\mathbf{V(i_s)}$	$\mathbf{V(i_s) \rightarrow V(i_d)}$
U	$c = any : U(i_s) \rightarrow U(i_d)$
C	$c = any : C(i_s) \rightarrow C(i_d)$

$\mathbf{V(i_s)}$	$\mathbf{V(i_s) \rightarrow V(i_d)}$	
S	$c = m : S(i_s) \rightarrow S(i_d)$	$c = b : S(i_s) \rightarrow D(i_d)$
	$c = hlp : S(i_s) \rightarrow PS(i_d)$	$c = hrt : S(i_s) \rightarrow PD(i_d)$
PS	$c = m, hlp : PS(i_s) \rightarrow PS(i_d)$	$c = b, hrt : PS(i_s) \rightarrow PD(i_d)$
PD	$c = m, hlp : PD(i_s) \rightarrow PD(i_d)$	$c = b, hrt : PD(i_s) \rightarrow PS(i_d)$
D	$c = m : D(i_s) \rightarrow D(i_d)$	$c = b, hrt : D(i_s) \rightarrow PS(i_d)$
	$c = hlp : D(i_s) \rightarrow PD(i_d)$	
$v \in V$	$c = u : v(i_s) \rightarrow U(i_d)$	

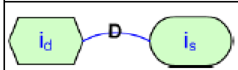
Backward Contribution



$\mathbf{V(i_d)}$	$\mathbf{V(i_d) \rightarrow V(i_1) \ldots V(i_n)}$
U	$U(i_d) \rightarrow \bigvee_{j=1}^{n} U(i_j)$

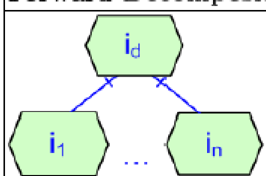
$\mathbf{V(i_d)}$	$\mathbf{V(i_d) \rightarrow V(i_1) \ldots V(i_n)}$
S, PS	$PS(i_d) \rightarrow (\text{for } r_j^c \in \text{Pos}, \bigvee_{j=1}^{n} PS(i_j) \vee \text{for } r_j^c \in \text{Neg}, \bigvee_{j=1}^{n} PD(i_j))$
C	$C(i_d) \rightarrow \Big(\bigvee_{j=1}^{n} C(i_j) \vee (\text{for } r_j^c \in \text{Pos}, \bigvee_{j=1}^{n} PS(i_j) \wedge \text{for } r_j^c \in \text{Neg}, \bigvee_{j=1}^{n} PS(i_j)) \vee (\text{for } r_j^c \in \text{Pos}, \bigvee_{j=1}^{n} PD(i_j) \wedge \text{for } r_j^c \in \text{Neg}, \bigvee_{j=1}^{n} PD(i_j))\Big)$
D, PD	$PD(i_d) \rightarrow (\text{for } r_j^c \in \text{Pos}, \bigvee_{j=1}^{n} PD(i_j) \vee \text{for } r_j^c \in \text{Neg}, \bigvee_{j=1}^{n} PS(i_j))$

Forward Dependency



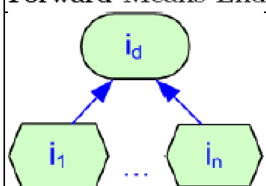
$\mathbf{V(i_s)}$	$\mathbf{V(i_s) \rightarrow V(i_d)}$
$v \in V$	$v(i_s) \rightarrow v(i_d)$

Forward Decomposition



$\mathbf{V(i_d)}$	$\mathbf{V(i_1) \ldots V(i_n) \rightarrow V(i_d)}$
S	$(\bigwedge_{j=1}^{n} S(i_j)) \rightarrow S(i_d)$
PS	$(\bigwedge_{j=1}^{n} PS(i_j)) \rightarrow PS(i_d)$
U	$((\bigvee_{j=1}^{n} U(i_j)) \wedge (\bigwedge_{k=1}^{j} PS(i_k) \wedge \bigwedge_{p=j+1}^{n} PS(i_p)) \rightarrow U(i_d)$
C	$((\bigvee_{j=1}^{n} C(i_j)) \wedge (\bigwedge_{k=1}^{j} \neg PD(i_k) \wedge \bigwedge_{p=j+1}^{n} \neg PD(i_p)) \rightarrow C(i_d)$
PD	$(\bigvee_{j=1}^{n} PD(i_j)) \rightarrow PD(i_d)$
D	$(\bigvee_{j=1}^{n} D(i_j)) \rightarrow D(i_d)$

Forward Means-Ends



$\mathbf{V(i_d)}$	$\mathbf{V(i_1) \ldots V(i_n) \rightarrow V(i_d)}$
S	$(\bigvee_{j=1}^{n} S(i_j)) \rightarrow S(i_d)$
PS	$(\bigvee_{j=1}^{n} PS(i_j)) \rightarrow PS(i_d)$
U	$((\bigvee_{j=1}^{n} U(i_j)) \wedge (\bigwedge_{k=1}^{j} \neg PS(i_k) \wedge \bigwedge_{p=j+1}^{n} \neg PS(i_p)) \rightarrow U(i_d)$
C	$((\bigvee_{j=1}^{n} C(i_j)) \wedge (\bigwedge_{k=1}^{j} PD(i_k) \wedge \bigwedge_{p=j+1}^{n} PD(i_p)) \rightarrow C(i_d)$
PD	$(\bigwedge_{j=1}^{n} PD(i_j)) \rightarrow PD(i_d)$
D	$(\bigwedge_{j=1}^{n} D(i_j)) \rightarrow D(i_d)$

Table 58 Forward and Backward Propagation Axioms

7.4.5 Human Judgment

We have provided a formal definition for the need for human judgment over intentions in Section 6.4.8. When human judgment is required for an intention, given a target evaluation value for the recipient intention, *target(i)*, the user is asked the following question:

"Results indicate that i must have a value of target(i).

Enter a combination of evaluation labels for intentions contributing to i which would result in target(i) for i.

$(\forall j; j = 1 \ldots n, r_j: i_j \rightarrow i)$

i_j , r^c_j, *(choice of one of S, PS, U, C, PD, D)*

…"

We provide a screenshot of the dialog window for the backward question over our simple example model in Figure 35. More details concerning implementation of the backward procedure can be found in Chapter 11.

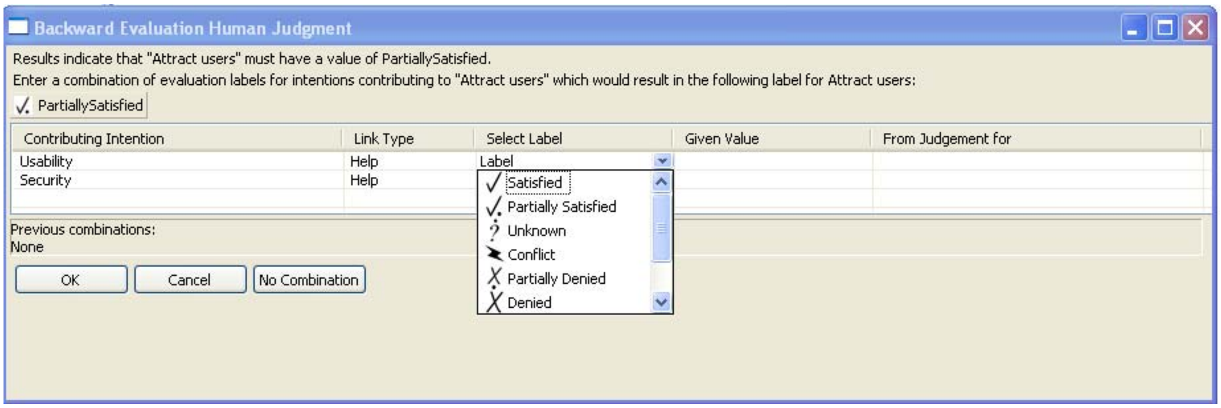


Figure 35 Screenshot of Human Judgment Dialog for Backward Analysis

When a judgment is provided by the user, the SAT formula is adjusted as follows:

- Forward and backward axioms in the SAT formula which propagate to or from *i* are removed. These are axioms of the form:
  - $(Any\ combination\ of\ v(i_1)\ \ldots\ v(i_n),\ v \in V) \rightarrow v(i)$
  - $v(i) \rightarrow (Any\ combination\ of\ v(i_1)\ \ldots v(i_n),\ v \in V)$
- New axioms representing the human judgment are added, for each r_j, r^c_j: $i_j \rightarrow i$, the value provided by the user for i_j: $v_j(i_j) \in \mathcal{V}$, is added to a forward and backward axiom as follows:
  - Forward: $(v_1(i_1) \wedge \ldots \wedge v_n(i_n)) \rightarrow target(i)$
  - Backward: $target(i) \rightarrow (v_1(i_1) \wedge \ldots \wedge v_n(i_n))$

In addition, we when encoding human judgment, we add the constraint that i must not have a conflict, to avoid situations where the SAT solver will assign extra values to i. For example if target(i) = PS(i), then the following would be added to Φ:

$$\neg U(i) \wedge \neg C(i) \wedge \neg PD(i)$$

7.4.6 Backward Analysis Algorithm

Simplified Java code implementing the backward algorithm can be found in Figure 36. Generally, the algorithm converts the model to CNF form, using the components of formula described in the previous sections (lines 7 and 8 in Figure 36). Two versions are converted, one using both the forward and backward propagation axioms to try to find a solution, `cnf`, and one using only the backward axioms in order to find targets for intentions, `cnfBack`. In a loop which terminates when no more intentions require human judgment (lines 9, 14, 15), the algorithm calls zChaff to find a solution for `cnf` (line 10). If a solution is found (line 11), the algorithm displays the non-conflicting results (line 13) and finds the topmost (closest to a root) intentions which need human judgment (line 13, 16). The target for each of these intentions is found by running the solver on `cnfBack` (line 17, 18) and taking the maximum label result for each intention, using the ordering in 4.

For each topmost intention needing human judgment, the user is prompted for judgment (line 21), and the judgment is added to the forward and backward `cnf` as described in Section 3.3 (line 23, 24). If the user provided some judgments, the list of topmost intentions needing human judgment is added to a stack (line 27). If, in the main loop, zChaff cannot find a solution (line 28), zMinimal is used to find the minimum core, which is displayed to the users (line 29-31). In this case, or when the user has no more judgments to add (line 25, 26), the algorithm backtracks, popping the last set of intentions needing human judgment from the stack (line 26) and backtracking over the `cnf` and `cnfBack` formula (removing the judgment axioms and adding back in the default forward and backward propagation axioms) (line 38, 39). Control is returned to the main loop (line 9) where the process starts again by finding a solution for the `cnf` (line 10). Only judgments over intentions in the minimal core are re-asked when backtracking (not shown in Figure 36). If the procedure backtracks, but there are no more intentions to backtrack over, the algorithm ends with no result (line 41-43).

```java
Dimacs cnf; Dimacs cnfBack;
zChaffSolver solver = new zChaffSolver();
zMinimalSolver minSolver = new zMinimalSolver();
ModeltoAxiomsConverter converter = new ModeltoAxiomsConverter(model);
Stack<Vector<Intention>> hjStack = new Stack<Vector<Intention>>();

void reason() {
   cnf = converter.convertBothDirections();
   cnfBack = converter.convertBackward();
   while( true )  {
      int result = solver.solve(cnf);
      if (result == 1) {
         HashMap<Intention, int[]> results = solver.getResults();
         Vector<Intention> needHJ = findHJAndDisplayResults(results);
         if (needHJ.size() == 0) {  //answer found, no judgments needed
            showMessage("Success!");  return; }
         Vector<Intention> topMostHumanJudgment = findTopMost(needHJ);
         solver.solve(cnfBack);  //used for intermediate targets
         Hashmap<Intention, int[]> backResults = solver.getResults();
         int hjCount = 0;
         for (Itention i: topMostHumanJudgment) {
            if (promptForHumanJudgment(i, backResults.get(i))) {
               hjCount++;  //count number of judgments given
               cnf = converer.addHumanJudgment(cnf, i);
               cnfBack = converter.addHumanJudgment(cnfBack, i); } }
         if (hjCount == 0) {  //user has no more hj to add
            if (backtrack() == -1) { return; }  }
         else { hjStack.push(topMostHumanJudgment); } }
      else if (result == 0) {  //solver found no solution
         minSolver.solve(cnf); //find unsat core
         String minResults = minSolver.getResults();
         showMessage ("Backtracking: " + minResults);
         if (backtrack() == -1) { return; } } } }

int backtrack() {
   if (hjStack.size() > 0) {  //there are judgments to backtrack over
      Vector<Intention> needHJ = hjStack.pop();
      for (Intention i: needHJ) {  //backtrack over of the last judgments
         cnf = converter.backtrackHumanJudgment(cnf, i);
         cnfBack = converter.backtrackHumanJudgment(cnfBack, i);  }
      return 1;
   } else { //there are no judgments to backtrack over
      showMessage("Target(s) unsatisfiable.  Ending.");
      return -1;  }  }
```

Figure 36 Simplified Java Code for the Backward Analysis Algorithm

7.4.7 Example

To illustrate the algorithm, we run an example over the model in Figure 34.

Iteration 1: The SAT solver is run on the cnf SAT formula. A satisfying assignment is found; however, there are intentions which need human judgment: Attract Users and Usability, of which Attract Users is the topmost. We prompt for human judgment, asking the users what combination would produce a partially satisfied value for Attract Users. The users indicate that Usability and Security must be partially satisfied. cnf and cnfBack are modified accordingly and the procedure loops.

Iteration 2: The SAT solver is called again on the new cnf. Human judgment is still needed, and the procedure asks the user for input on the conflicted intention nearest to the root, Usability. The user indicates that for Usability to be partially satisfied Ask for Secret Question should be satisfied and Restrict Structure of Password should be denied. The SAT formulas are modified to reflect this information.

Iteration 3: The solver is run on the new cnf. In this case, the formula is unsatisfiable, if Restrict Structure of the Password is denied then Security is denied, when the rule collected in the first iteration indicates it must be partially satisfied in order for Attract Users to be partially satisfied. The procedure backtracks (modifying the cnf encodings) and the user is then asked for more possible viable combinations for the last point of judgment, Usability. No more possibilities are given which would make Usability partially satisfied. The procedure backtracks again and asks the user if there are more combinations of source intentions that would produce a partially satisfied value for Attract Users. This time the user indicates that if Security were satisfied and Usability had a conflict value, Attract Users would be partially satisfied. The axioms to and from Attract Users are again removed and the human judgment axioms are added.

Iteration 4: The solver is run again on the modified cnf. Usability requires human judgment. The user indicates that for Usability to have a conflict value, Restrict Structure of Password and Ask for Secret Question can be satisfied. The encodings are updated.

Iteration 5: The solver is run on the new cnf. This time, not only is satisfying assignment found, but all intentions in the model do not require human judgment. The procedure finishes, informing the user that in order for Attract Users to be partially satisfied, Restrict Structure of Password and Ask for Secret Question must be satisfied

7.4.8 Run Time, Termination, Soundness, and Completeness

Run Time: In analyzing the runtime we exclude a detailed exploration of the runtime complexity of zChaff or zMinimal, marking these values as (zChaff) and (zMinimal). In practice, the running time of SAT approaches would be effected by the number of Means-Ends (OR) decompositions and multiple incoming contribution links, as each of these structures provide further labeling alternatives, increasing the search space.

The main loop reason() in Figure 36 will loop until hjCount == 0. In the worst case each iteration involves a single new judgment for every intention. If a model has n intentions ($|\mathcal{I}|$) and each intention has a maximum of q sources (q incoming links), there is a maximum of $6^q \times n$ possible judgments, where $q < n$. The run time of the initial axiom conversion is $6l$, where l is number of links in the model ($|\mathcal{R}|$). The cost of adding or backtracking human judgment on the converter is also l (finding the right axiom by links). In addition, the worst case runtime of findHJAndDisplayResults and is n, findTopMost is $2n$, and backtrack is $2nl$. If zChaff returns a result, the worst case runtime is either $2ln + 3n +$ (zChaff) or $2nl$, else, without a result, it is $2nl$ + (zMinimal). Assuming (zMiminal) ≈ (zChaff), the worse case runtime for reason() is then $6^q n(2ln + 3n +$ (zChaff)$) + 6l$, or $O(6^q(ln^2 + n$(zChaff)$))$. Although this is an exponential value, q is usually a small number, less than 5 or 6. Also, although there is a worst case of 6^q possible combinations of human judgment for each intention, only a small subset of these judgments will be acceptable for the user, who will try to maximize positive contributions.

We have applied our implementation of the procedure to several example models, with the automated portion of the procedure completing within seconds. Running times over example models are reported in Chapter 11. As the procedure is meant to be used over models created by hand, the maximum size of such models is reasonably constrained by human cognition. Potential procedure efficiency improvements are discussed in the next section.

Termination: If the user continues to make the same judgments, the procedure will not terminate. However, the current implementation provides a list of previous judgments attempted which did not produce a solution. As there are a finite number of intentions each with a finite number of sources, there are a finite number of human judgments which can be provided (6^q). If the user does not continually reuse judgments, the procedure terminates.

Soundness: Our axiomatization is sound if the axioms correctly capture forward and backward propagation. Reviewing the propagation rules in Table 58, along with the description of forward propagation in Section 6.4.6, will show that the formal propagation rules reflect the intended interpretation of the model reflected by the prose description of propagation.

Completeness: Our axiomatization is complete if it covers all propagation listed in our interpretation of agent-goal model propagation. An examination of Table 58 will show that we have considered propagation rules for every combination of evaluation label and link type, given the restrictions on our model in Section 7.4.2. Currently the forward procedure takes into account additional agent-goal model structures such as mixes of link types and cycles, as a result, the backward procedure is not entirely complete when compared to the forward procedure. However, we claim it is sufficiently complete to produce useful results in terms of our requirements for early analysis of agent-goal models. We explore this claim further through case studies in Chapter 12. Future expansions of the backward procedure could aim to increase its completeness.

7.5 Conclusions, Discussion, and Future Work

In this chapter, we have introduced an interactive, iterative procedure for backward analysis of agent-goal models for early RE analysis. The procedure has addressed several of the questions and challenges listed in Section 1. It poses a specific type of question to the user ("What source values could produce a target value?") during iterations where conflicts or unaddressed partial values are detected, modifying the encoding by adding and removing axioms. We have defined assumptions concerning backward propagation over human judgment situations which include explicit conflict and unknown values and which avoid over constraining the model. The run time of the procedure has been analyzed, and although the worst case is exponential over the maximum number of children in the model, in practice this number is small (R1). We judge this goal to have a conflict label (previously partially denied). Further examples in Chapter 11 will test the scalability on models of a practical size. This procedure complements the forward procedure in Chapter 6, expanding the interactive (R5, R10) analytical power (R13) of agent-goal models, encouraging stakeholder involvement in the early modeling and analysis process and increasing the likelihood of system success. We summarize the contributions of this Chapter in Figure 37.

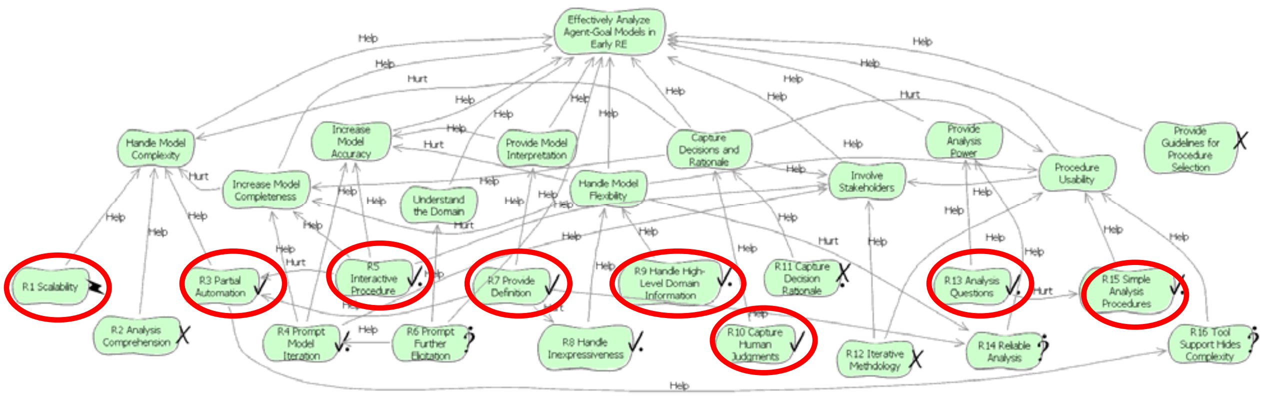


Figure 37 Contributions of Chapter 7 to the Requirements for analysis of Agent-Goal Models in Early RE

Advancement over existing work. In (Giorgini et al., 2004b), the authors present a formal framework allowing for backward reasoning with goal models. The results for each goal are presented using two values, one for satisfaction and one for denial. Often results contain many goals which are both partially satisfied and denied, making it challenging to derive an overall analysis conclusion. The backwards procedure described in this work could be seen as an expansion or modification of this procedure, as we have borrowed our general CNF formulation and part of our analysis predicates from this work. However, we make several expansions and modifications to the procedure in (Giorgini et al., 2004b), as follows:

- Incorporating user interaction through human judgment, allowing users to resolve conflicts and make tradeoffs.
- Accounting for additional agent-goal syntax (dependency, unknown, and some+/- links).
- Accounting for additional analysis values (conflict, unknown).
- Producing results which have only one value per intention.
- Providing information on model conflicts when a solution cannot be found.

Use of SAT: In the early stages of this work we considered encoding agent-goal model propagation as a Constraint Satisfaction Problem (CSP) or Satisfiability Modulo Theories (SMT) Problem. However, in order to capture the presence of conflicts and the need for human judgment, each intention would have to be assigned multiple variables, making the encoding roughly as complex as our SAT encoding. Consideration was also given to the use of an incremental SAT solver, reusing the state-space when clauses are added to the encoding.

However, as our algorithm not only adds, but removes and re-adds clauses, these types of algorithms could not be applied.

Model Restrictions: In order to facilitate procedure termination and simpler axiomatization, we have adopted the model restrictions from (Giorgini et al., 2004b). In some cases, the procedure will still terminate, even if the model contains a cycle. However, we cannot guarantee termination for all models with cycles. Mixing link types, for example, a dependency and a means-ends or decomposition link with the same target, will not cause the procedure to hang. However, we have not added axioms to describe the semantic interpretation and behavior of analysis in these cases. Currently, the implementation and axioms would add all incoming labels from multiple types of relations to the target intention. To be consistent with the forward procedures, the results of each individual link type should be combined using an AND relation (Section 6.4.9). Future work can add axioms to Table 58 and modify the CNF encoding to change this behavior, making it consistent with the forward procedure.

Future Procedure Optimizations: The backward algorithm can be optimized in several ways. The algorithm in Figure 36 backtracks by removing and adding clauses from the CNF encoding and recalling the SAT solver. Instead, it could store the zChaff solver results in another stack, popping those results when backtracking, reducing the number of times zChaff is called on average. The number of human judgment situations could be reduced in practice by optionally reusing judgments within a single evaluation, across both forward and backward evaluation, and by deriving judgments from existing judgments. However, automation of judgment reuse may deny users the opportunity to reconsider their judgments. Currently we favor user flexibility and choose to not automatically reuse previous judgments in analysis, although previous judgments are displayed as part of the judgment question, and can be viewed in a tree (see Chapter 11 for more implementation details).

Chapter 8 A Methodology for Agent-Goal Model Creation and Analysis

In order to facilitate the use of agent-goal models for early RE analysis, we provide a set of guidelines for elicitation and scoping, model creation, iteration, and analysis. Case study experience has led to the belief that a highly specific methodology for creating and analyzing agent-goal models may be too restrictive, due to varying characteristics of the domain and available modelers. As a result, we advocate this methodology as only a general guide, or a series of suggestions. Depending on the context, the role of stakeholders, and the specific required outcome of the modeling process, the methodology can be adapted as needed.

Our experience with modeling has shown that the process of modeling and analysis is as important, perhaps even more important, for understanding and discovery as the resulting models. Ideally, this approach would be applied in cooperation with domain representatives. This allows representatives to have a sense of ownership over the model and the decisions made as a result of the modeling process, as described by Stirna & Persson (2007). However, it may be difficult to acquire stakeholder buy-in to the modeling process, and in these cases analysts can undertake the modeling process using other sources, including interviews, documents and observations.

In this Chapter, we introduce a sample methodology using this procedure to guide users through the process of modeling and evaluation. The methodology is divided into two parts, model creation and analysis. The first part guides initial elicitation, scoping and modeling. The suggested modeling process starts with the recording of actors and associations, then dependency

This chapter includes excerpts from the following papers/reports:

Horkoff, J., & Yu, E. (2009a). Evaluating Goal Achievement in Enterprise Modeling – An Interactive Procedure and Experiences. The Practice of Enterprise Modeling (pp. 145-160). Springer.

Horkoff, J., & Yu, E. (2010b). Interactive Analysis of Agent-Goal Models in Enterprise Modeling. International Journal of Information System Modeling and Design (IJISMD). IGI Global.

links, then actor intentions and rationale links. Model creation is followed by a process which uses both forward and backward analysis, showing modelers how to start the iterative, interactive analysis process over agent-goal models. The first two sections of the analysis section are meant to act as "sanity checks" in the model, checking that it produced sensible answers for a variety of questions, while the last section is intended to support more useful analysis in the domain.

Although the suggested methodology is described in two parts, which are further divided into steps and sections in sequence, the method is meant to be iterative, and flexible. Each step may bring forth ideas which contribute to areas of the model created in previous steps. The analysis section, in particular, is aimed at prompting beneficial model iteration. If the methodology is followed without the direct participation of stakeholders, each stage may result in questions which should be answered by domain experts. This knowledge should be incorporated back into the model at any stage.

Referring to our requirements for analysis of agent-goal models in early RE introduced in Chapter 3, the contents of this chapter directly address R12, Iterative Methodology (Figure 38). Indirectly, the suggested method addressed R4, Prompt Model Iteration, by providing a series of steps which are intended to lead to beneficial model changes. Variations of this methodology have been applied in industrial and individual studies, as described in Chapter 12.

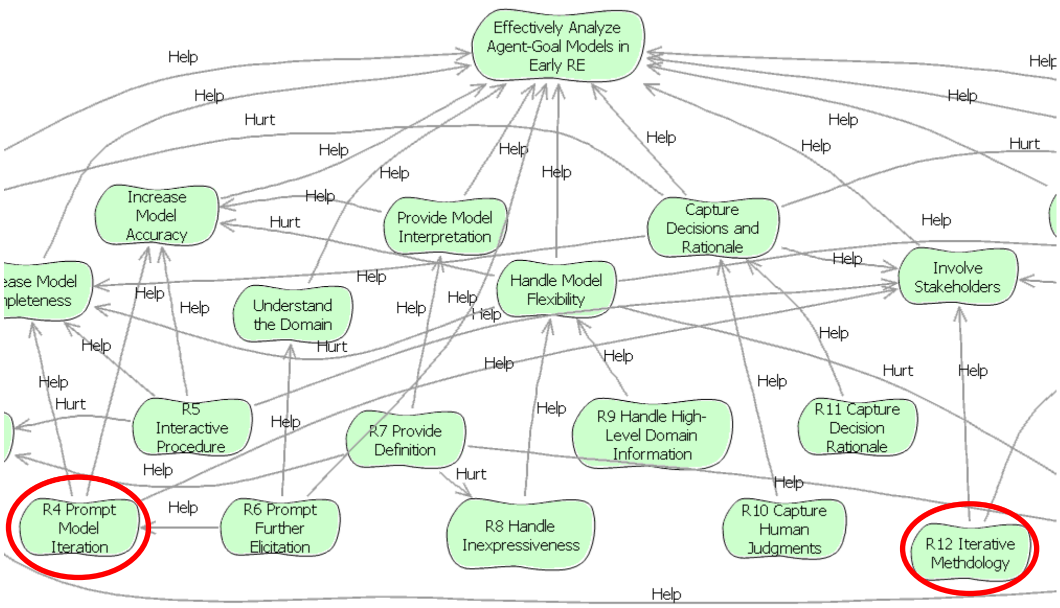


Figure 38 Focus of Current Chapter in the Requirements for analysis of Agent-Goal Models in Early RE

8.1 Analysis as Part of Model Creation

In this section we provide a suggested methodology for elicitation and construction of an i* model, including analysis as a high-level step in this process. The analysis step will be described in more detail in the following section. We will illustrate the model creation method using a simplified example from the first phase of the counseling service case study, as described in several previous chapters. The entire case will be described in more detail in Chapter 12.

Apply the following steps iteratively:

Stage 1: Purpose and Elicitation

Identify scope or purpose of the modeling process. It is important to identify one or more issues of focus for the modeling process. This determines the scope of the analysis in each of the modeling steps, continually questioning the relevance of including certain actors, dependencies and intentions. Focusing on a specific topic or area helps to manage scalability of the resulting models.

Example: In the social service example, the purpose of the first phase of the study was to identify and evaluate the effectiveness of various technical alternatives for providing online youth counseling. As such, the models focused on the organizations use of technology or systems interfacing with the internet, and on those individuals in the organization who used such systems, or who directed the overall vision for the organization.

Identify modeling participants and/or model sources. As stated, ideally the models would be created along with selected domain stakeholders who would act as a source for the information captured in the models. Alternatively, if stakeholders are not directly available, interviews, documents, observations or other sources can be used. These sources could also be used to supplement the knowledge of any participating stakeholders.

Example: In the example, stakeholders were generally unfamiliar with modeling as a tool for analysis and had difficulty committing significant amounts of time. As a result, models were developed by the analysts using stakeholder interviews and information gained through site visits. We interviewed a variety of stakeholders interfacing with the online counseling service, including counselors, IT staff and various levels of management.

Stage 2: Model Creation

Identify relevant actors and associations. With the model scope in mind, identify relevant domain actors and the relationships between them. This could include specific stakeholders or more abstract roles or organizations. Helpful analysis questions include: "Who is involved?" and "How are they related?"

Example: The actual case study identified 63 relevant actors. In our simplified example we focus on youth, counselors, and the counseling organization.

Identify relevant dependencies. In the same or a separate model, identify the dependencies between actors. Helpful analysis questions include: "Who needs what?" and "What do they provide in return?"

Example: The actual case study identified 405 potentially relevant dependencies, a subset of these dependencies are depicted in the SD model in Figure 17, repeated below in Figure 39.

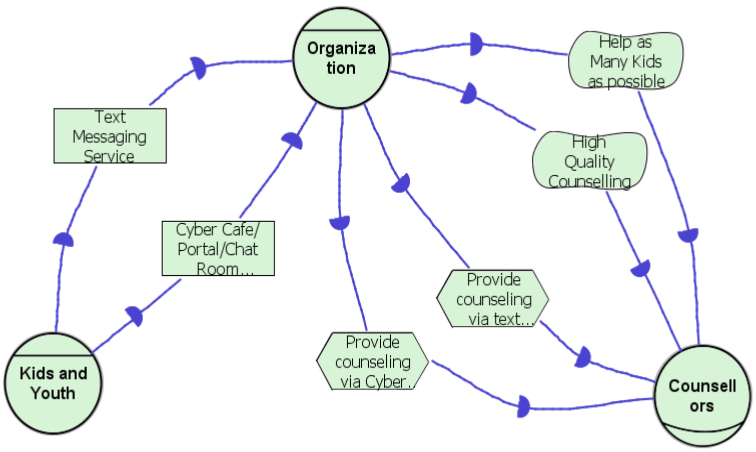


Figure 39: Simplified SD Model for Youth Counseling

Identify actor intentions. This stage is divided into three iterative sub-steps:

> **Identify actor intentions.** Using the sources, identify what actors want, what tasks they perform, how they achieve things.
>
> **Match dependencies to actor intentions.** Using the dependencies found in the previous stage, answer "why?" and "how?" questions for each dependency, linking all dependencies to existing or new intentions within an actor.
>
> **Identify relationships between intentions.** Identify how the actor intentions relate to each other, whether it is through a functional AND/OR hierarchy or through positive or negative contributions. New intentions may be discovered. Ideally, no intentions should be isolated.

Example: A subset of the intentional intentions identified in the case study has been shown in Figure 23 (repeated in Figure 16, Figure 11, and Figure 2). Even for this simplified example, a complex web of contributions and dependencies are formed.

Stage 3: Analysis

Evaluate alternatives within the model. Apply the evaluation procedures introduced in Chapter 6 and Chapter 7 to the model. This stage is described in more detail in the next section.

Example: In the counseling service case study, several online counseling alternatives such as moderated forums, chats, email, and text messaging were analyzed and compared using the evaluation procedure. Specific example evaluations for the counseling service and other case studies have been presented in Chapter 6 and Chapter 7 as a means to illustrate the evaluation procedures.

8.2 Suggested Analysis Steps

We have argued in this thesis that the utility of i* models can be increased by using models for analysis, i.e. using models to answer interesting questions over the domain. However, studies have shown (Chapter 12) that it can be difficult to know how to start the analysis process. In this section, we provide some initial analysis questions meant to provide some baseline analysis results and to test the utility of the model. We then provide some guidance for finding domain-driven questions to apply to the model. If the results of the analysis at any point do not make sense in light of the domain, the analysis and modeler should make changes to the structure or content of the model.

The first application of the model typically involves evaluating the most obvious alternative, and often helps to test the "sanity" of the model. Isolated intentions which do not receive an evaluation value can be identified. Evaluation results which are not sensible can either reveal a problem in the model or an interesting discovery concerning the domain. Changes prompted by the evaluation results should be made in the model.

As the model evolves, more complicated or less obvious questions or alternatives can be analyzed. Further model changes can be made. The process continues until all viable alternatives are analyzed, an alternative has been selected, or a sufficient knowledge of the domain has been gained, depending on the initial purpose of the modeling process determined in Step 1.

We will demonstrate our methodology using an example model describing the General Chair of the ICSE (International Conference on Software Engineering). We select this model as it has a higher degree of complexity than our previous counseling service example, and therefore better demonstrates the selection of leaf and root intentions and the complexity of analysis alternative selection. A description of this study is included in our validation chapter; however, it is not necessary to understand the details of the model to in order to understand the analysis methodology.

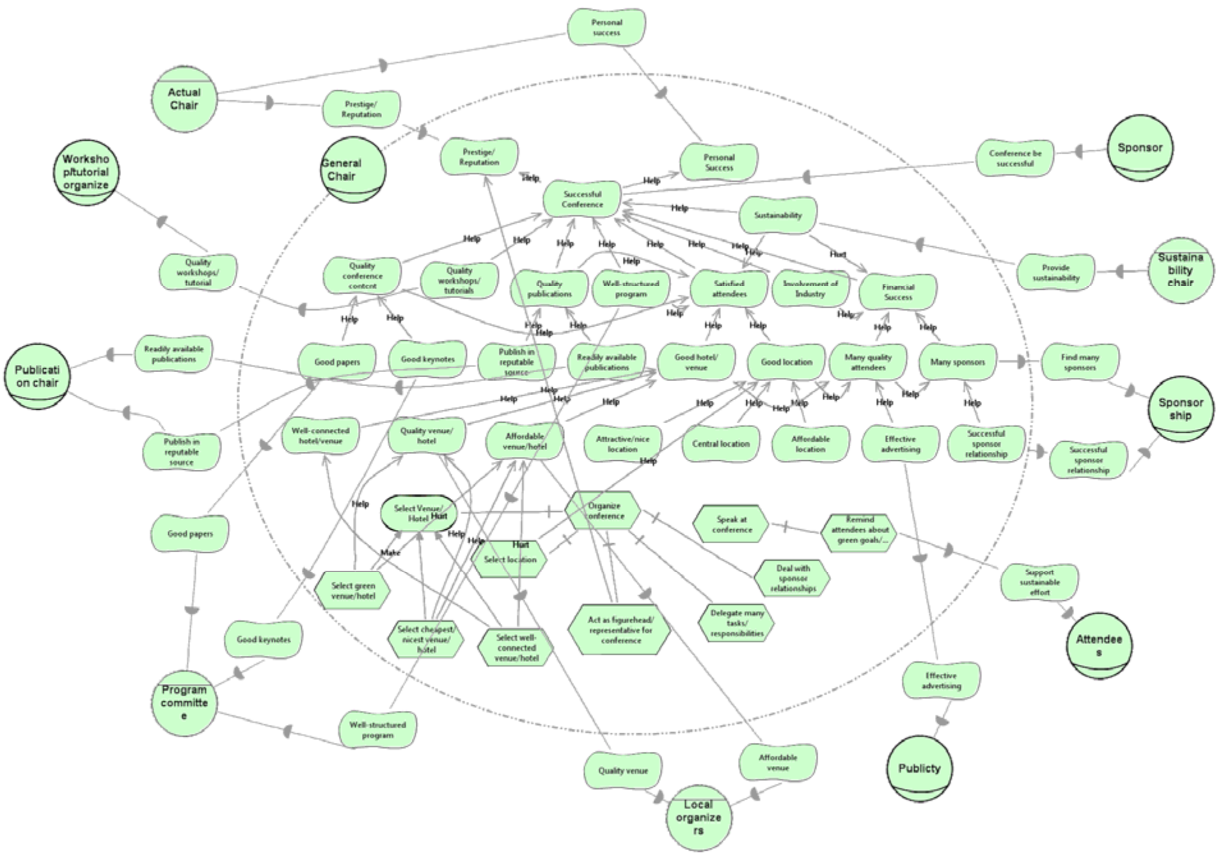


Figure 40 Model Showing the General Chair of the ICSE (International Conference on Software Engineering)

8.2.1 Initial Analysis Questions

The initial analysis questions can be divided into two types, those that analyze the affects of alternatives (forward analysis), and those that test the achievement possibilities in the models (backward analysis).

8.2.1.1 Alternative Effects

First, we provide some initial analysis steps which helps to determine the affects of alternatives on the content of the model, supporting "what if?" questions.

Identify the leaf intentions in the model. A formal definition of leaf intentions has been provided in Section 5.4. Typically these represent model alternatives which can be selected or not in possible configurations. In this way, alternatives can be turned "off" (denied) and "on" (satisfied). We can analyze the affects of choosing one or more sets of likely alternatives.

Note: Some alternatives may not be leaf intentions; they may be located at a higher-level in the model. Such intentions identified by the modelers can be considered as alternatives in step 2.

Note: In a sufficiently complete model leaf intentions typically represent implementation options. In the process of finding leaf intentions, you may find leaves which are more abstract. These intentions are potential candidates for further decomposition. Examples of this type are shown in Figure 41.

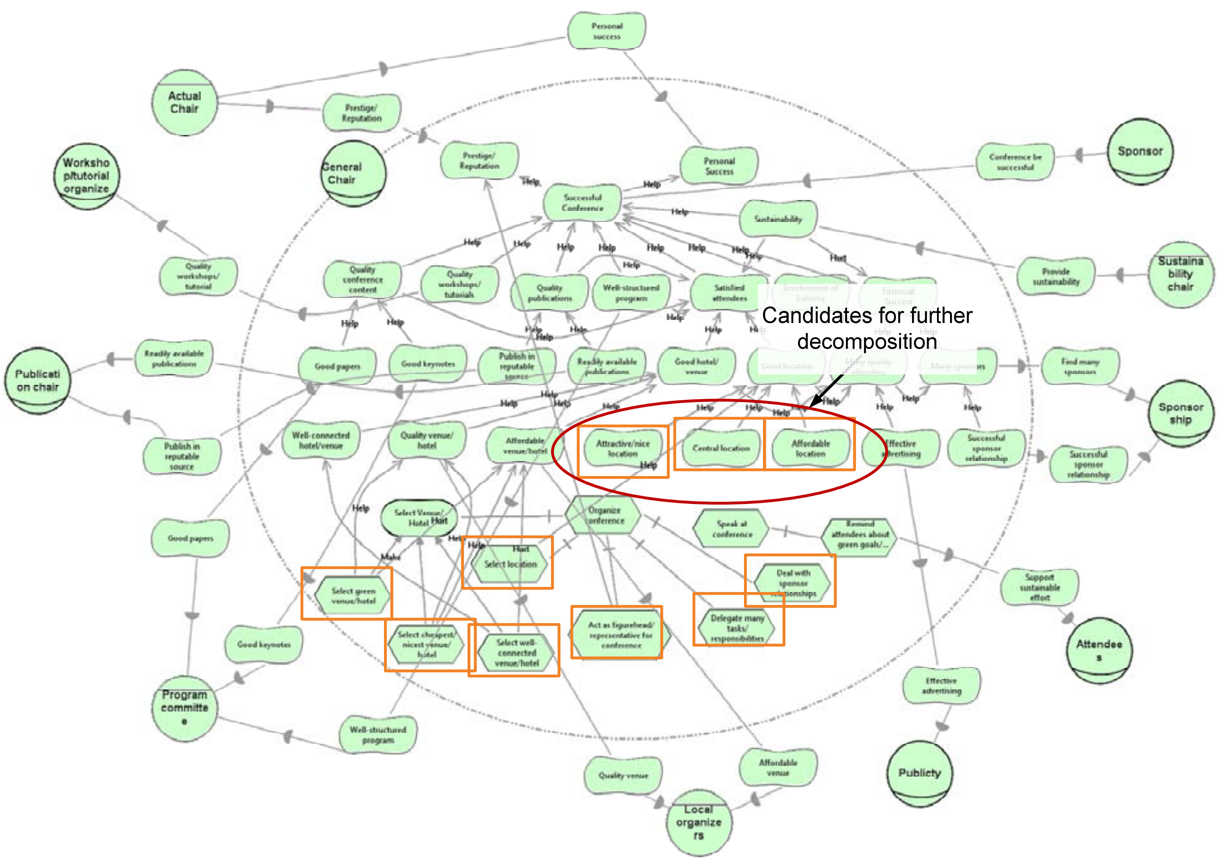


Figure 41 Model Showing the General Chair of the ICSE with Leaf Intentions Identified

Find and evaluate plausible combinations of alternatives. Select as many of the following which apply to your model:

Implement as much as possible. As a "most positive" baseline analysis alternative, analyze the affects of choosing (satisfying) all alternatives which are not mutually exclusive. If alternatives are exclusive (cannot be chosen at the same time), pick the most likely or current alternative. An example for the ICSE model is shown in Figure 42.

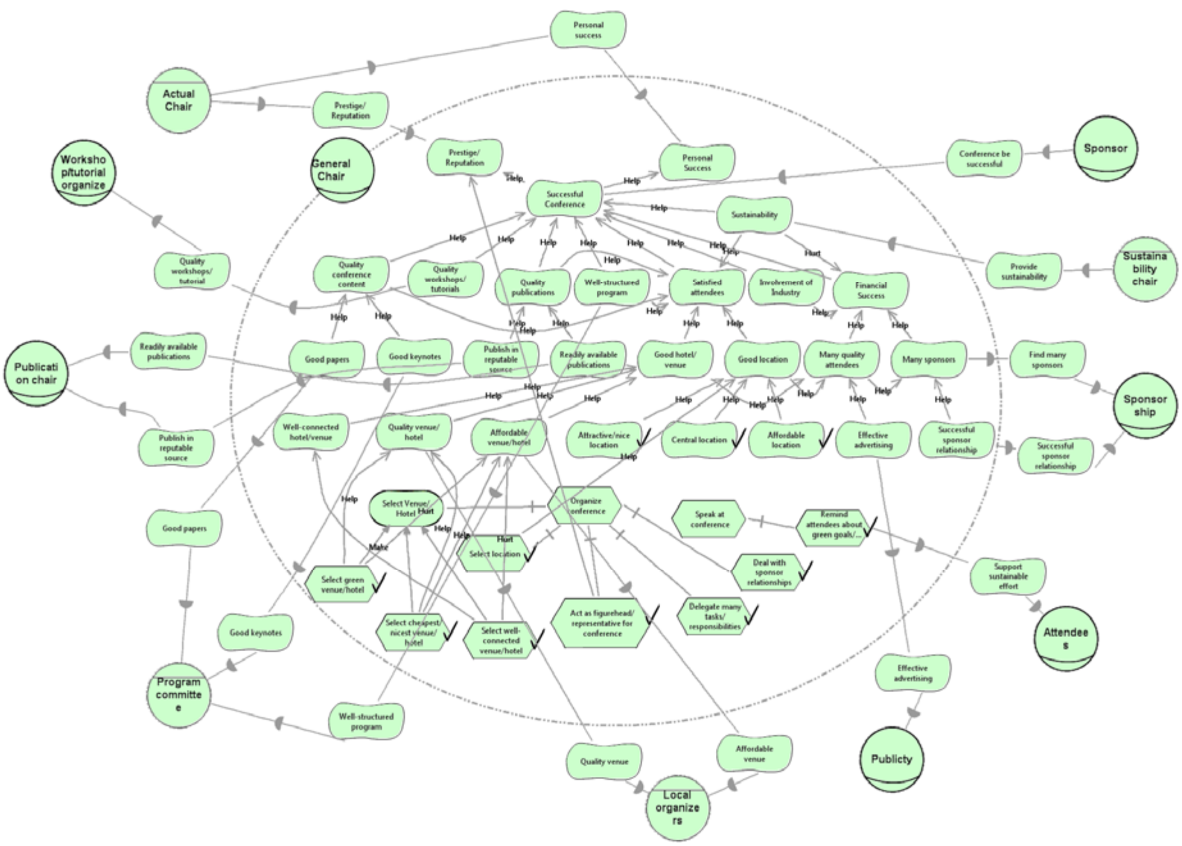


Figure 42 Model Showing the General Chair of the ICSE with Most Positive Baseline Alternative

Implement as little as possible: As a "most negative" baseline analysis alternative, analyze the affects of the not choosing (denying) all alternatives possible. If in some cases one alternative must be chosen, chose the least likely or most expensive alternative. An example for the ICSE model is shown in Figure 43.

Figure 43 Model Showing the General Chair of the ICSE with Most Negative Baseline Alternative

Reasonable implementation alternatives. Select analysis alternatives which seem likely or promising in the domain. Each combination of alternatives can be given a different descriptive name. An example for the ICSE model analyzing the selection of a well-connected hotel is shown in Figure 44.

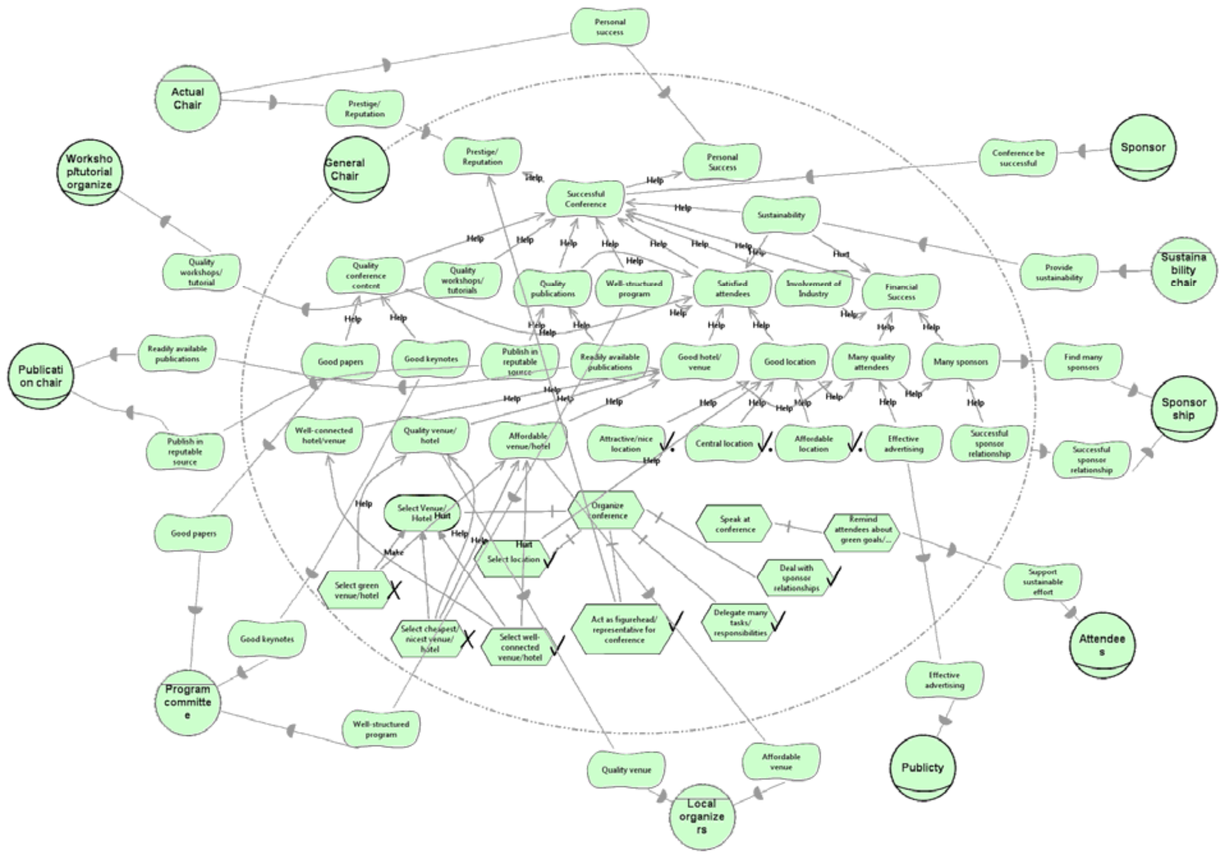


Figure 44 Model Showing the General Chair of the ICSE with Reasonable Analysis Alternative

8.2.1.2 Achievement Possibilities

Next, we provide some initial analysis steps which help to determine targets in the model, and if it is possible to achieve these targets, supporting “is this possible?” questions. If at any point in the process a solution cannot be found, consider whether this is a result of the structure and contents of the model (i.e. can the model be improved), or an inherent property (e.g., a conflict) of the domain.

Identify the root intentions in the model. Typically these represent the “top”, higher-level, or most important goals of various actors. Root intentions have been given a formal definition Section 5.4. These are often the goals that we want to achieve. Figure 45 highlights the root intentions for the ICSE example. We can assign target levels of satisfaction to these important intentions and try to determine whether it is possible to achieve these levels by selecting an alternative in the model.

Note: Some very important intentions may not be roots in the model; they may be located one or more levels from the top. Such intentions identified by the modelers can be considered root goals in this step.

Note: In a sufficiently complete model root intentions typically represent the higher-level or most important goals. In the process of finding root intentions, you may find roots which are at a lower-level or are more minor. These intentions are potential candidates for further "why?" brainstorming, finding higher-level motivating goals.

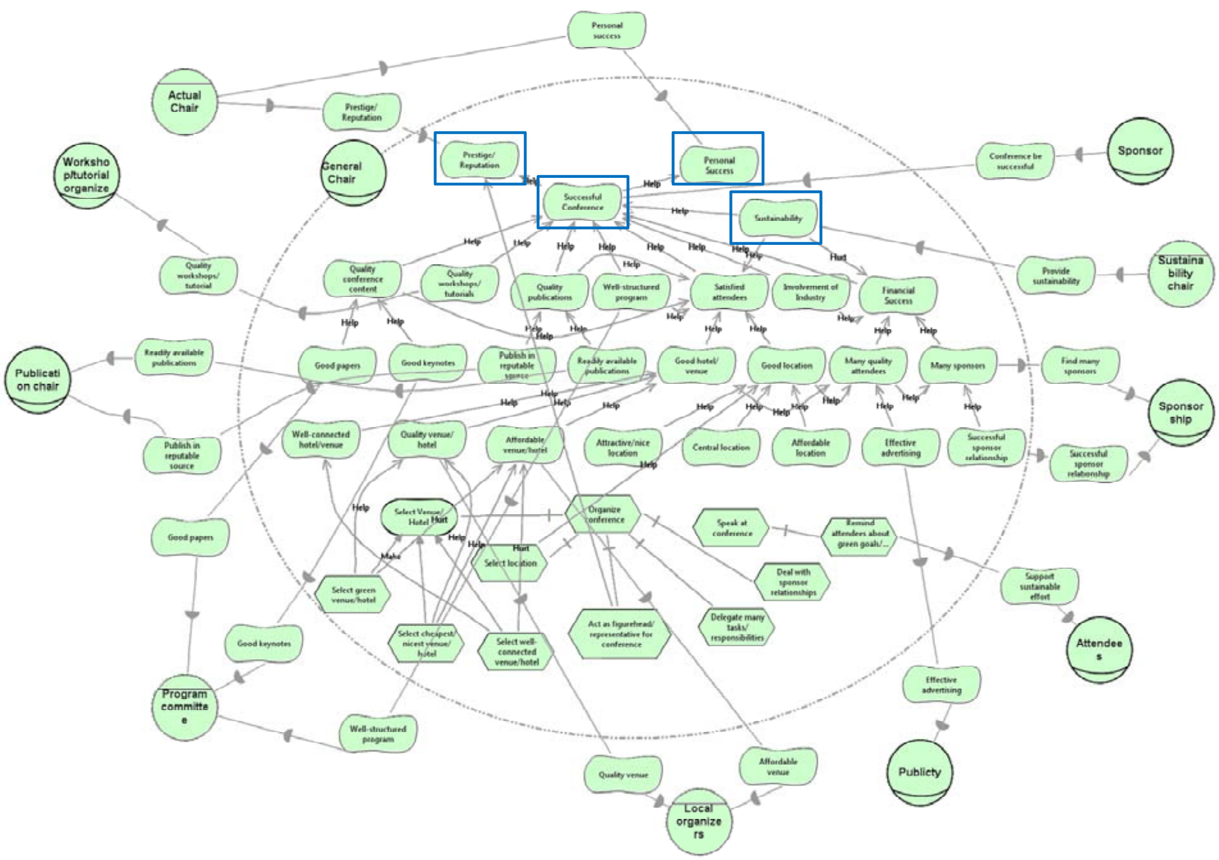


Figure 45 Model Showing the General Chair of the ICSE with Root Intentions Identified

Evaluate possible combinations of targets for top intentions. Select as many of the following which apply to your model:

Maximum Targets. Assign target levels of satisfaction to the top intentions in the model which reflect the maximum desired level of satisfaction. Typically this will involve all top intentions being fully satisfied; however this can be relaxed if it is already known that full satisfaction is not possible for all top goals. If it is possible to achieve maximum targets in the model, what alternative(s) produce these targets?

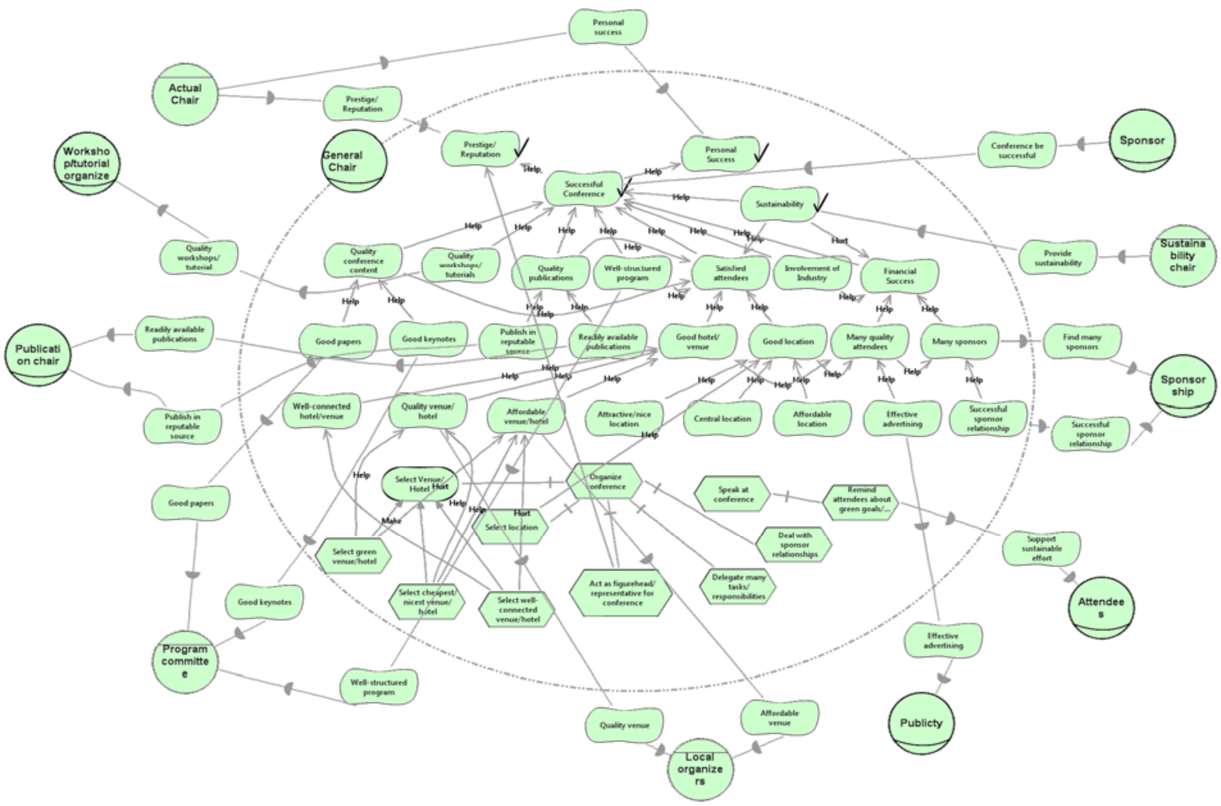


Figure 46 Model Showing the General Chair of the ICSE showing a Maximum Target

Minimum Targets: Assign target levels of satisfaction or denial to root intentions in the model which reflect the minimum level of satisfaction/denial that may be permissible. What are you willing to give up? What must you have? At what level of satisfaction? What alternatives produce these minimums? If an intention does not have to be at least partially satisfied, no target label should be placed. An example minimum target for the ICSE model is shown in Figure 47.

Note: There may be more than one combination of minimum targets, i.e., if we give up this intention, we must have this other intention instead.

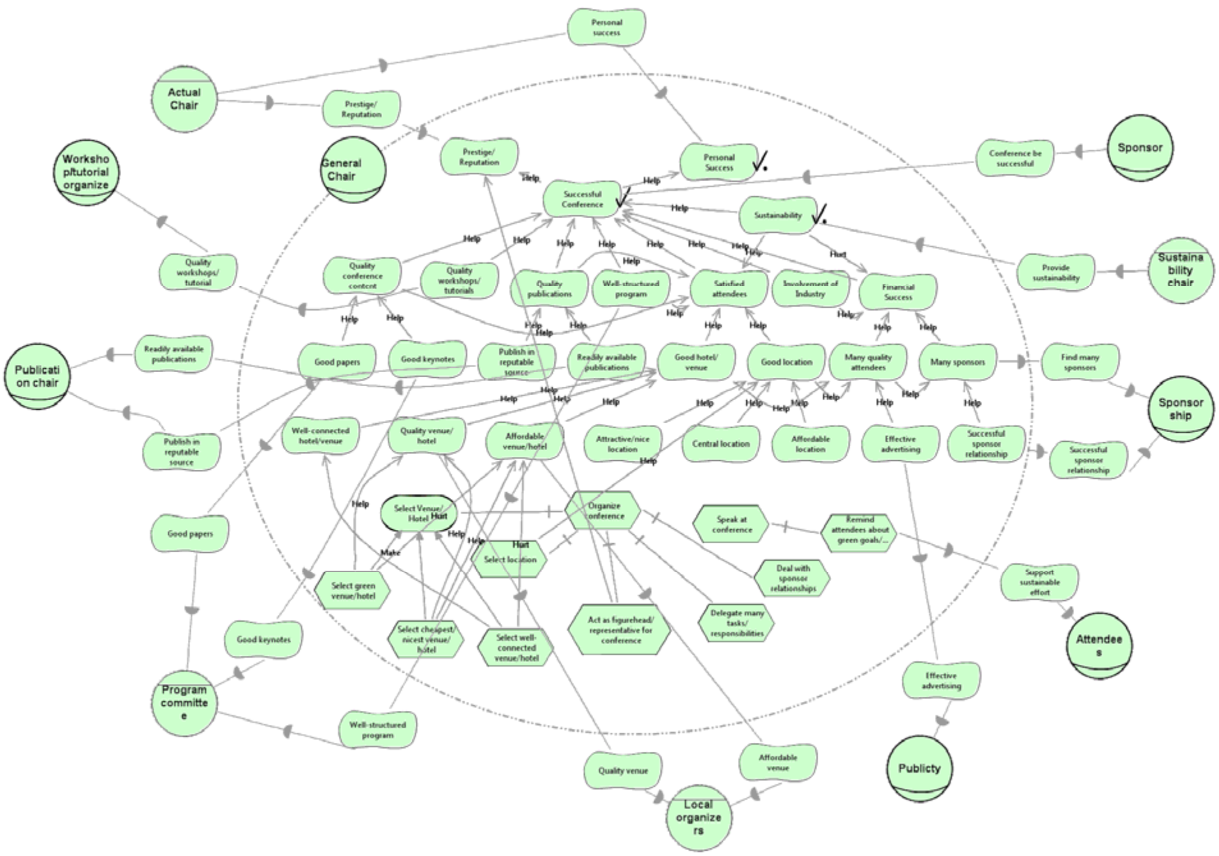


Figure 47 Model Showing the General Chair of the ICSE showing an initial Minimum Target

Iteration over minimum targets: The previous step has identified a minimum level of satisfaction for target intentions. If this step resulted in finding an alternative which achieved this minimum target, try gradually increasing the satisfaction level of top goals, each time checking feasibility within the model. If the previous step did not find an alternative which achieved the minimum target, try to further relax the desired target intentions. Search for the maximum level of satisfaction possible whilst selecting an alternative in the model. A final minimum target, relaxed from the target in Figure 47, is shown in Figure 48.

Note: There may be more than one optimal combination of targets for top intentions which are possible in the model.

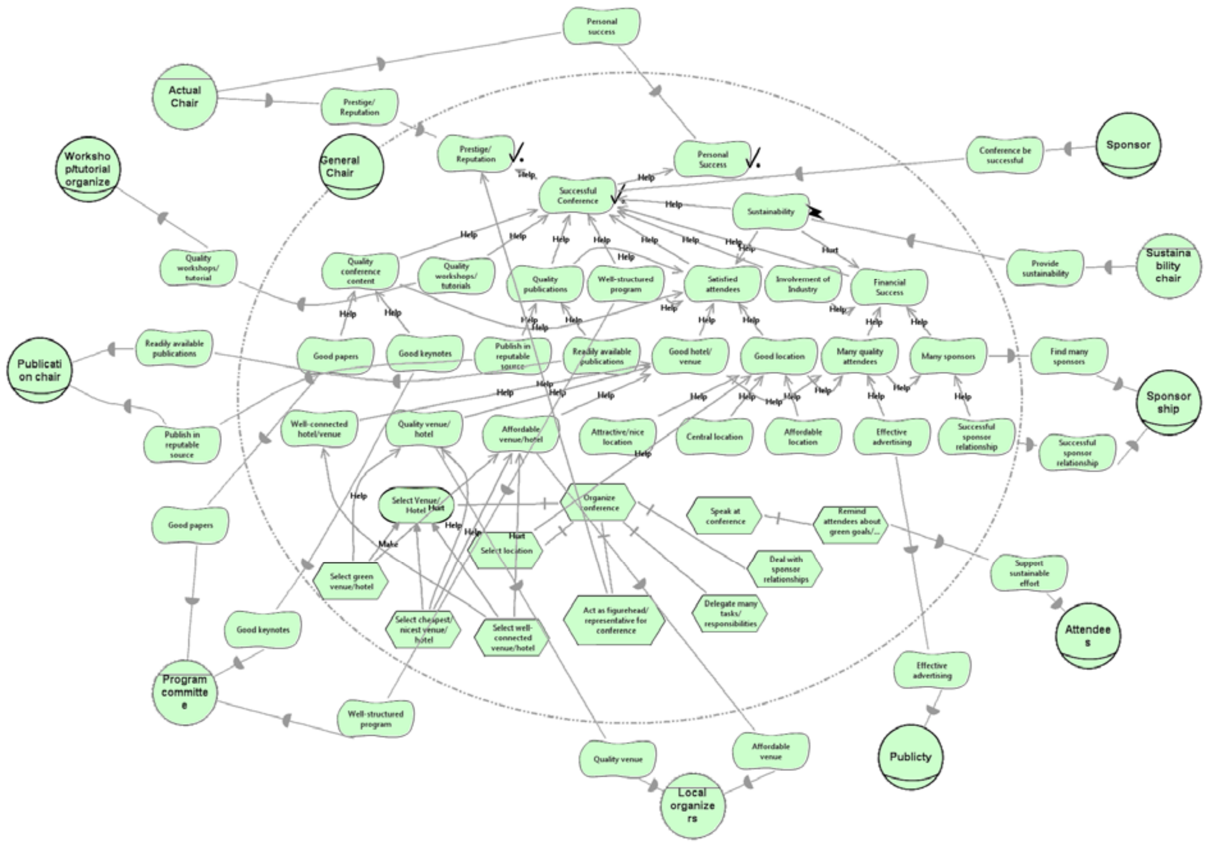


Figure 48 Model Showing the General Chair of the ICSE showing a Final Minimum Target

8.2.2 Domain-Driven Analysis

Once initial baseline analysis questions have been asked over the model, we can use the model to answer interesting or important questions in the domain. For example:

- What design options to select?
- Will a particular option work? For whom?
- Will the goals of a certain stakeholder be satisfied?
- Will a particular goal be satisfied?
- Can a set of particular goals be satisfied at the same time?

In the example model, questions could include:

- What options for organizing the conference should the general chair choose, not choose?
- What if the general chair selected a green hotel?
- Is it possible for the top goals of the general chair to be satisfied at once?
- Is it possible to fully satisfy sustainability? Sustainability and successful conference?

If the model is not able to help in answering these questions, ask:

- Is the model missing some important concept or relationship? Can this be added?
- Is the difficulty due to the limitation of i*modeling and analysis?

Although we make the claim that analysis over goal and agent-oriented models can be useful in many dimensions (see Section 1.5, for example), the approach does have several limitations, both in the modeling framework and analysis techniques. As mentioned in Section 3.1, an effective requirements process would make use of a variety of RE techniques, using or not using models. Other analysis techniques may be used when the limitations of qualitative i* analysis are reached. See Chapter 13 for a more detailed discussion of the limitations of the analysis framework introduced in this work.

8.3 Discussion and Conclusions

In this chapter, we have introduced a suggested methodology for model creation and analysis. The process is intended to be iterative, with each step potentially leading to changes in the artifact or results of previous steps.

When comparing our methodology to other existing methods for i* creation, (see Section 2.4 for a summary of such approaches), we have tried to describe an approach which is more flexible. As the steps involved in elicitation and analysis in early RE will vary according to domain factors (stakeholder expertise, available time, willingness to participate in modeling, nature of the domain), the steps in the methodology should also be variable. Analysts can use their knowledge of the domain and its participants, including the purpose of the modeling process, to rearrange or omit certain steps if applicable. Our approach has also emphasized iteration, especially as a result of model analysis, using initial analysis as a form of sanity checking for the model.

By introducing a methodology we have addressed R12, the need for an Iterative Methodology and have aimed to address R4, Prompt Model Iteration (Figure 49). Our claimed concerning R4 are tested more thoroughly with empirical evaluations in Chapter 12.

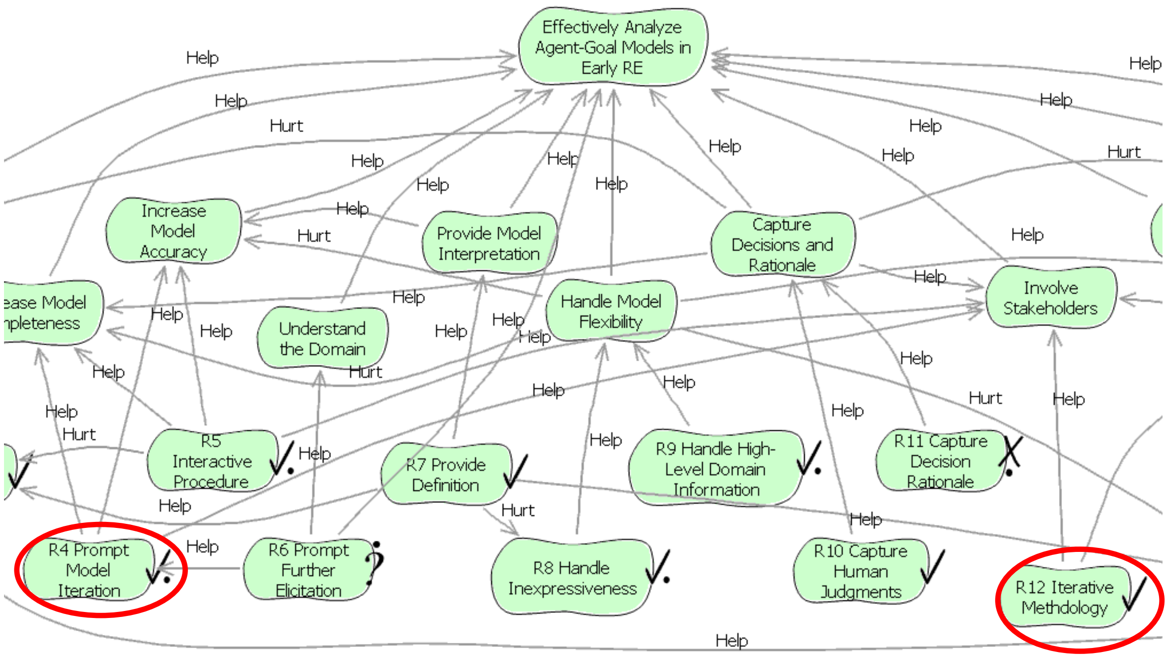


Figure 49 Focus of Current Chapter in the Requirements for analysis of Agent-Goal Models in Early RE

Chapter 9 Visualization Techniques for Iterative, Interactive Agent-Goal Model Analysis

While goals models can visually present alternate ways for achieving goals and how actors depend on each other, it can be challenging to follow the reasoning through complex paths in the model. In studies which tested the utility of procedures for guiding analysts to perform interactive forward and backward reasoning through i* models, we uncovered several difficulties faced by study participants. Specifically, users often have trouble choosing suitable starting points for analyzing the model, finding intentions over which they are providing judgments, and understanding conflicts among alternatives. The details of these studies will be described in Chapter 12. In this chapter we develop visualization mechanisms to alleviate these difficulties. Roots and leaves in the model are automatically detected and highlighted as suggested starting points for analysis. Intentions involved in a human judgment are highlighted. Goals within a conflicting path are highlighted during analysis. The first and third visualization mechanisms are tested with five follow-up studies, described in Chapter 12. The results suggest several further visualization mechanisms which could support analysis.

The visualizations described in this chapter are intended to work towards supporting the comprehension of analysis results (R2), provide a precise definition of concepts used (R7), are used as part of an iterative methodology (R12), and help to hide the complexity of the procedure through tool support (R16). We highlight these contributions in our summary diagram of the requirements for the early RE analysis of agent goal models, repeated in Figure 50.

This chapter is an expansion of the following papers/reports:

Horkoff, J., & Yu, E. (2010c). Visualizations to support interactive goal model analysis. Requirements Engineering Visualization REV 2010 Fifth International Workshop on (Vol. 70, p. 1–10). IEEE.

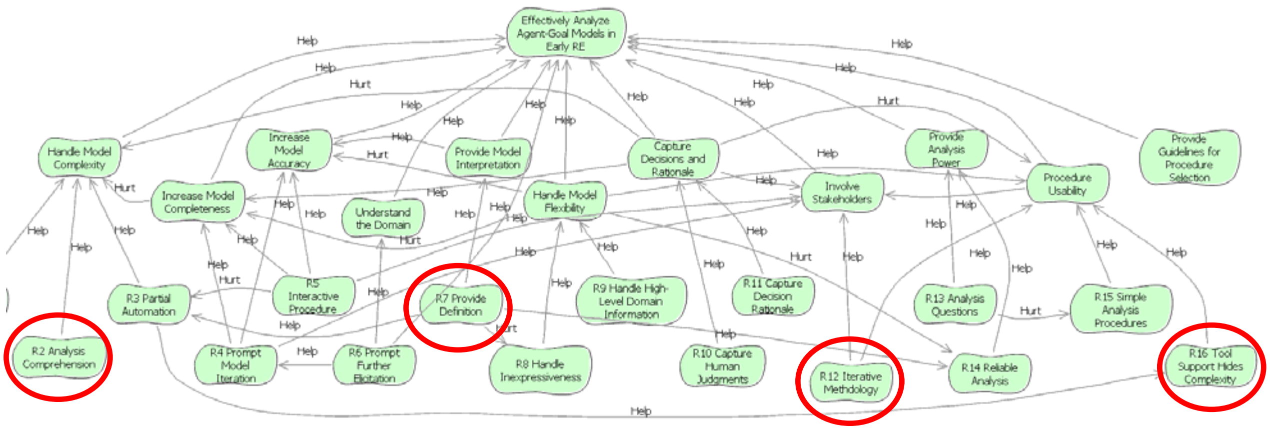


Figure 50 Focus of Current Chapter in the Requirements for analysis of Agent-Goal Models in Early RE

9.1 Motivation for Visualizations

We have introduced analysis procedures which are interactive, aiming to involve users in the RE process and improve the quality of the model, increasing domain knowledge. The interactive nature of the procedure means that visualizations are particularly important in helping users understand the model and analysis. In this work we introduce and study visual interventions aimed to aid users in interactive goal model analysis.

As part of our empirical evaluation of the analysis procedures introduced in this work, ten individual and one group case study were conducted. These studies aimed to test the effects of a guided interactive model analysis procedure, compared to ad-hoc (non-systematic) analysis. Qualitative analysis over the results revealed several challenges experienced by the users when attempting to interactively analyze a goal model. Although these issues may be addressed by increased training in i* and interactive analysis, enhanced visualization techniques can also help. In the current chapter, we focus on three issues which appeared prominently in those prior studies: starting analysis, finding intentions involved in judgments, and understanding model conflicts.

9.1.1 Starting Points for Analysis

Users had difficulty knowing where or how to start analysis in the models. In cases where they were given analysis questions to answer using the model, they had problems finding where to place the labels to reflect the analysis questions. In cases where they were encouraged to come up with their own analysis questions, they had difficulty knowing how to start analysis. In order to help participants, a suggested analysis methodology was developed (this methodology is now part of the description in Chapter 8). Based on experience from case studies described in Chapter 12, the methodology suggests to start forward analysis by identifying leaf intentions in the model and to start backward analysis by identifying root intentions. These intentions would serve as suggested starting points, although users are encouraged to add other starting points, or to make changes to the model if current model roots or leaves did not seem to be sensible analysis starting points.

However, results from the user studies reveal that users often have difficulty identifying leaves and roots in agent-oriented goal models. Four participants applied this methodology in the individual studies. All participants had at least some problems correctly identifying leaf or root intentions in the models. Although this seems like a simple task for users with training in the use of graphs as computational data structures, certain characteristics of i* models make this task challenging, namely:

- some i* links do not have an obvious direction (for e.g., dependency links),
- the presence of actors and actor boundaries can often cause users to ignore links coming into or out of an actor, and
- as i* models often contain both actors and cycles, they are typically not laid out in similar manner as a typical graph (roots on top, leaves on bottom).

Figure 51 shows a high-level view of an example i* model used in six of the individual case studies. Note the potential difficulty in identifying model leaves and roots. In this chapter, we provide a feature which automatically highlights model leaves and/or roots.

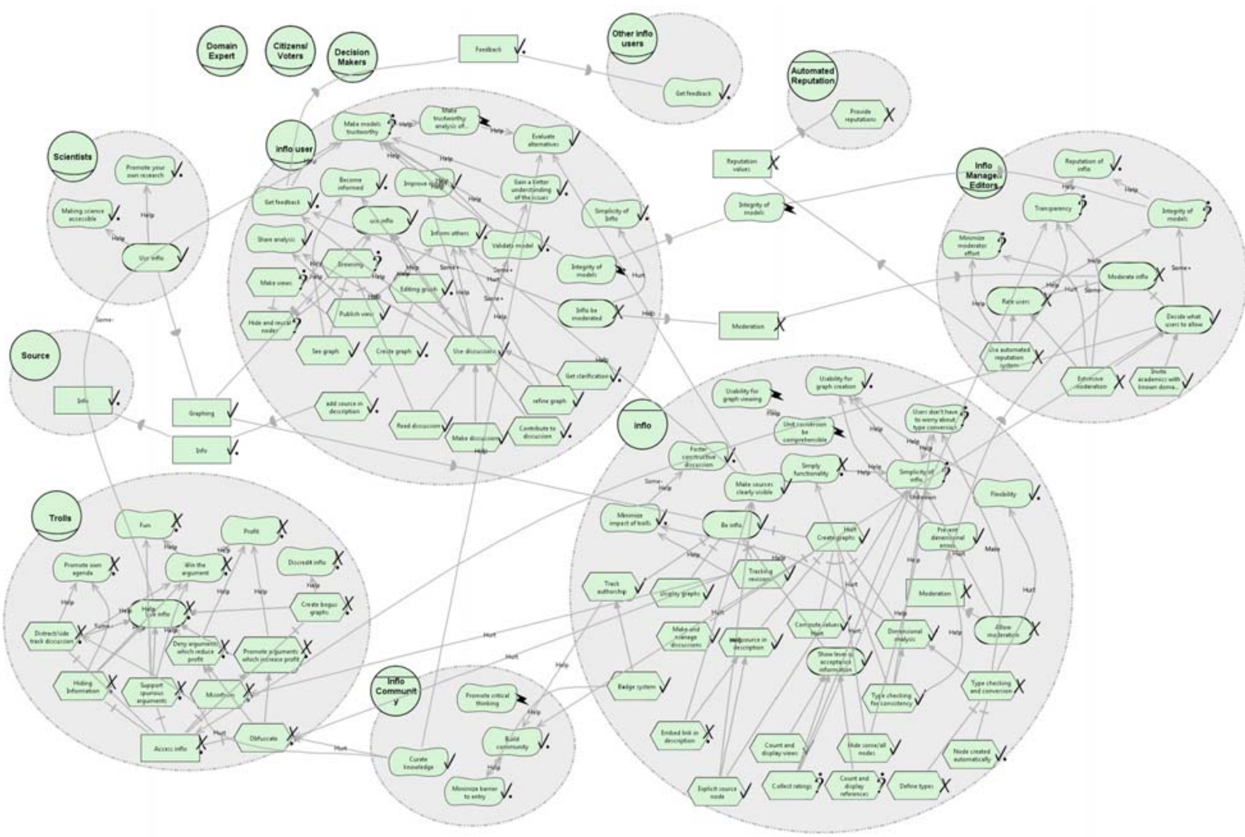


Figure 51 High-level view of the i* Model Resulting from the inflo Case Study

9.1.2 Finding Intentions Involved in Human Judgments

During interactive forward and backward analysis, when the users were asked to make judgments over an intention, they had difficulty finding that intention, and the intentions contributing to it, in the model. Users found it useful to find these intentions and examining their location and relation to other model elements. Sometimes these intentions are not located near each other, in some cases modelers draw contribution links across actors, and contributing intentions can be located on the other side of the canvas. For example, in the inflo model above (Figure 51) a backward analysis asking if Make models trustworthy can be partially satisfied produces a judgment shown in Figure 52. The reader can see the complexity in finding these intentions within the model. In this chapter, we highlight intentions involved in a human judgment.

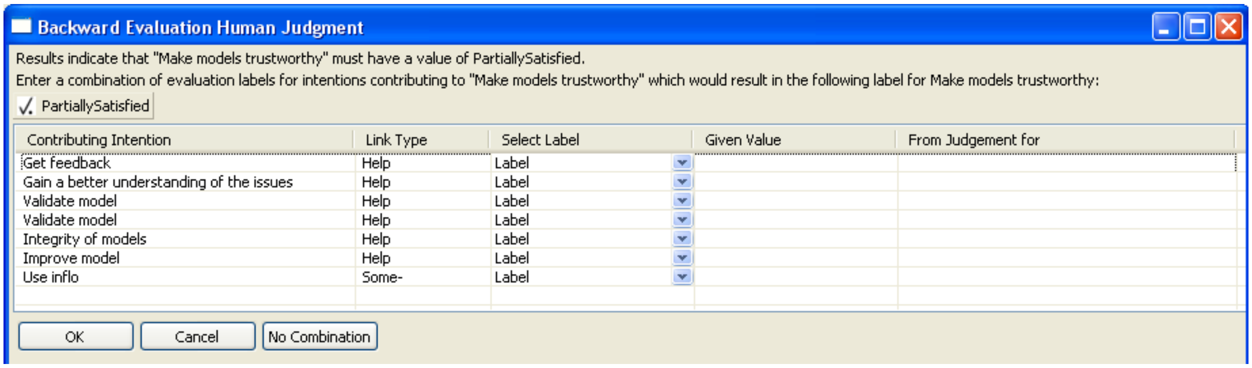


Figure 52 Example Backward Human Judgment Question for Figure 51

9.1.3 Understanding Conflicts

When the interactive backward procedure revealed conflicts in the models (inability to find a permissible assignment of analysis labels), users had difficulty understanding the reasons behind the conflicts. In the backward analysis direction, the interactive procedure described in Chapter 7 finds answers to analysis questions by translating the model into a satisfiability (SAT) problem. When model conflicts are discovered, it finds information about these conflicts by finding the UNSATisfiable core – a list of SAT clauses which are involved in the conflict. Such information would potentially guide their decisions while backtracking over human judgment thus far. During the studies, the user was presented with a list of intentions involved in the conflict. Results showed that this was insufficient to understand where in the model the conflict occurred, or why. Presenting this information to the user in a form which is understandable to users presents a challenge. In this chapter, we use the information in the UNSAT core to find model elements in the conflict path and elements which are the "sources" of the conflict, highlighting both types of elements in the model.

For example, a conflict occurred during the analysis of Figure 51. This figure shows the state of the model at the time of the conflict. The user is provided with text output listing the intentions involved in the conflict, in this case: Simple functionality, Type checking for consistency, Flexibility, Type checking and conversion, use inflo, Graphing, Be inflo, Create graphs, Node created automatically, Define types, Dimensional analysis, Usability for graph creation, and Simplicity of inflo. In this particular case two leaf intentions, Dimensional analysis and Type checking and conversion are both partially satisfied and not partially satisfied (PS and not PS), causing a logical conflict. The reader can see that even with the ability to zoom in and out to explore the model, locating the intentions and understanding the causes of the model conflict is difficult.

An alternative form of outputting the information from the conflict would be a list of the clauses involved the UNSAT core, as in **Figure 53**. However, we do not wish to assume that i* analysis users have a background in logic. In the next section we explore visualizations to improve the comprehensibility of conflicts.

The following intention clauses are conflicting:
not PS(Simple functionality) OR PD(Type checking for consistency)
not PS(Flexibility) OR PD(Type checking and conversion)
not PS(use inflo) OR PS(Graphing)
not PS(Graphing) OR PS(Be inflo)
not PS(Be inflo) OR PS(Create graphs)
not PS(Type checking and conversion) OR PS(Node created automatically)
not PS(Type checking and conversion) OR PS(Define types)
not PD(Type checking and conversion) OR PD(Node created automatically) OR PD(Define types)
not PS(Create graphs) OR PS(Dimensional analysis)
not PS(Dimensional analysis) OR PS(Type checking for consistency) OR PS(Type checking and conversion)
S(use inflo)
PS(Usability for graph creation)
not S(use inflo) OR PS(use inflo)
not PD(Define types) OR not PS(Define types)
not PD(Type checking for consistency) OR not PS(Type checking for consistency)
not PD(Node created automatically) OR not PS(Node created automatically)
not PS(Usability for graph creation) OR PS(Simplicity of inflo)
not PS(Usability for graph creation) OR PS(Flexibility)

Figure 53 Example Clauses in the UNSAT core for a Conflict during Backward Analysis in the Inflo Model in Figure 51

9.2 Visualizations to Support i* Analysis

In this section we provide technical details of three visualizations supporting the analysis usability issues described in the previous section.

9.2.1 Leaf and Root Intention Highlighting

In order to address difficulties in starting analysis, we have added the ability to automatically highlight leaf and root intentions in the model. We have provided a formal definition of leaf and root intentions in Section 5.4.1, repeated below:

Definition: leaf or root intention. *An intention* $i \in \mathcal{I}$ *is a leaf if there does not exist any relation,* $r \in \mathcal{R}$ *such that* $r : \mathcal{I} \rightarrow i$ *or* $r : \mathcal{I} \times \ldots \times \mathcal{I} \rightarrow i$*, it is a root if there does not exist any relation,* $r \in \mathcal{R}$ *such that* $r : i \rightarrow \mathcal{I}$ *or* $r : i \times \ldots \times \mathcal{I} \rightarrow \mathcal{I}$.

In our OpenOME implementation, described in more detail in Chapter 11, the user can select the "Mark Model Leaves" or "Mark Model Roots" options. Quick access buttons for each option appear in the toolbar (Figure 54). In each option root and leaf intentions appear blue or bright green, respectively. Figure 55 shows a conference planning model used in one of the individual studies (Chapter 12) with leaves and roots highlighted.

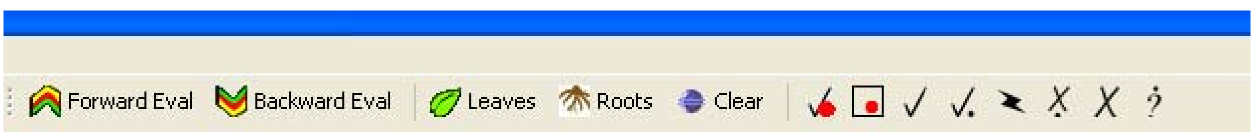


Figure 54 Screenshot of OpenOME Toolbar Showing quick Access to Leaf and Root Highlighting

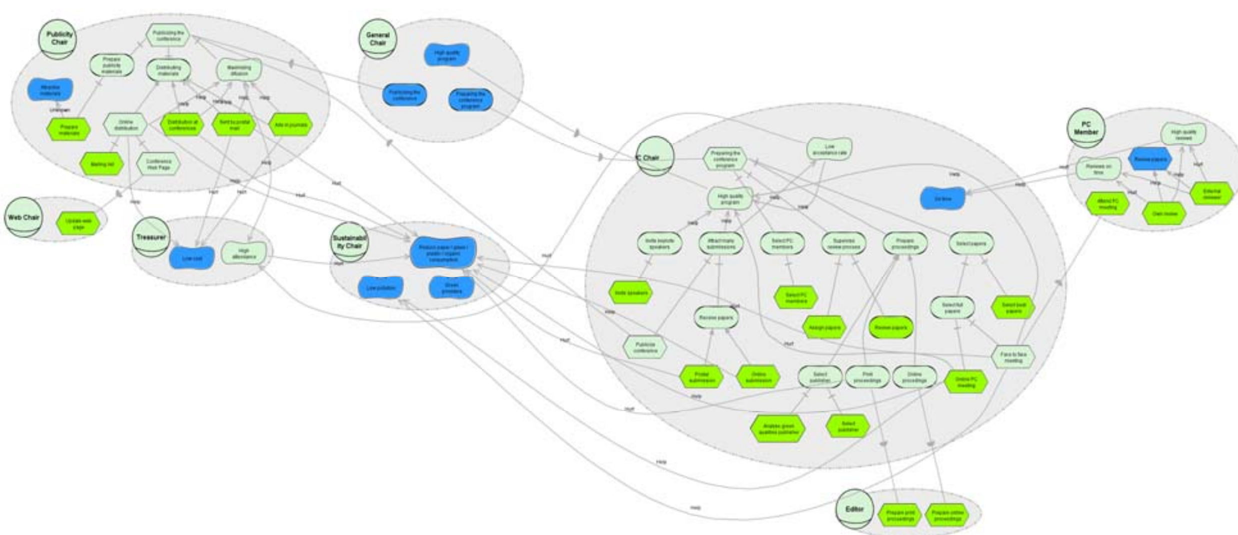


Figure 55 Model used in an Individual Study with Roots and Leaves Highlighted

9.2.2 Highlighting Intentions Involved in Judgments

We have added automatic highlighting of intentions involved in a human judgment situation to our analysis procedures. When the user is prompted for judgment in the forward or backward procedure, all the intentions involved in the judgment are highlighted in yellow. See an example for the inflo model in Figure 56. We have provided a definition of forward and backward propagation axioms in Section 7.4.4. In Section 6.4.8, we have described when human judgment is needed for an intention, $i \in \mathcal{I}$. Intentions involved in a judgment are simply the intentions present in the forward or backward propagation axioms leading to or from an intention over which judgment is needed. More specifically:

Definition: intentions involved in a human judgment. *If an intention, $i \in \mathcal{I}$, needs human judgment as defined in Section 6.4.8, the intentions in the propagation axioms leading to (forward analysis: (Any combination of $v(i_1)$ ... $v(i_n)$, $v \in \mathcal{V}$) $\rightarrow$ $v(i)$) or from (backward analysis: $v(i) \rightarrow$ (Any combination of $v(i_1)$ ...$v(i_n)$, $v \in \mathcal{V}$)), i_1 ... i_n, as well as i, are involved in a human judgment.*

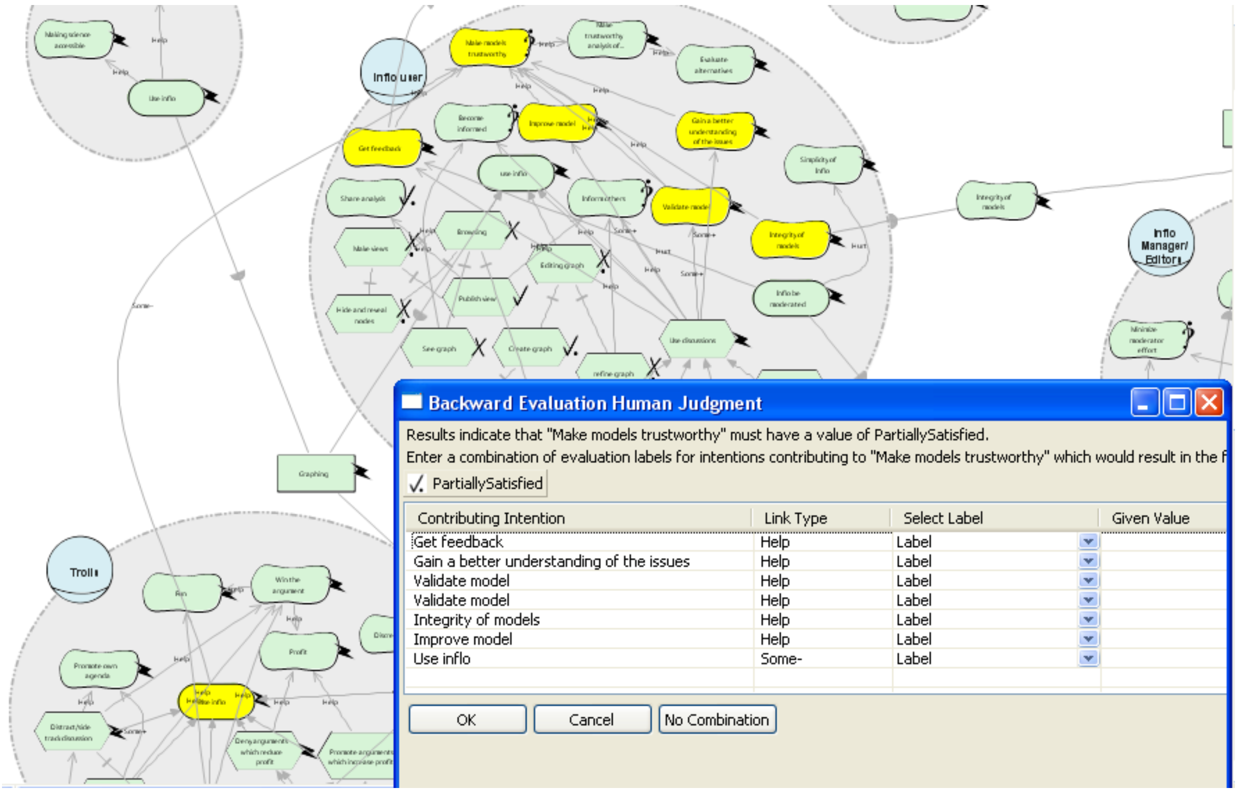


Figure 56 Conference Planning Example Showing Highlighting of Judgments Involved in Intentions

9.2.3 Conflict Highlighting

Using the information provided by the UNSAT core during a conflict, we highlight intentions in the model involved in the model conflict. The visualization differentiates between intentions on the path involved in the conflict, "root" intentions involved in the conflict, and intentions which were the "logical source" for the highlight, i.e. for which an analysis predicate is both true and not true (for e.g., $PS(i)$ and $\neg PS(i)$, $i \in \mathcal{I}$). We formally define these concepts in the remainder of this section. We start by defining a model conflict and an UNSAT core, and then continue by describing what it means for an intention to be involved in the conflict, for a clause to be a root clause in the conflict, or for an intention to be a logical source for a conflict.

Definition: model conflict. *When the SAT solver used in the backward analysis procedure cannot find a solution over a CNF representing a model, there is a conflict between the structure of the model and the consequences of the conflicts placed over the model. Specifically, in all* *possible assignment of variables, for one or more intentions, $i \in \mathcal{I}$, both v(i) and ¬v(i) hold, v(i) $\in \mathcal{V}$.*

Definition: UNSAT core from backward analysis. *An UNSAT core is an unsatisfiable subset of clauses, $c \in \mathcal{C}$, in an unsatisfiable CNF representing a model. Each clause c is a disjunction of one or more analysis predicates over intentions, v(i), for $v \in \mathcal{V}$ and $i \in \mathcal{I}$.*

For example, **Figure 53** provides an UNSAT core for an example conflict in Figure 51. More information concerning the construction of the CNF formalism from the structure of the model can be found in Chapter 7.

Definition: intention involved in conflict. *An intention, $i \in \mathcal{I}$, is involved in the conflict when it appears in one of the clauses, $c \in \mathcal{C}$, in the UNSAT core.*

For example, in the UNSAT core in **Figure 53** Simplicity of inflo and Graphing are intentions involved in the conflict.

Definition: root clauses in UNSAT core. *Clauses $c \in \mathcal{C}$ for which only one predicate, v(i), $v \in \mathcal{V}$, $i \in \mathcal{I}$, holds. These clauses are a subset of the initial target labels, ΦTarget, for backward analysis.*

For example, in Fig. 3, Usability for graph creation and Use inflo are root clauses.

Definition: logical source of conflict. *An intention, $i \in \mathcal{I}$ is a logical source of a conflict when:*

- *It is involved in the conflict, and*
- *An analysis predict is both true and not true for an intention, according to the UNSAT core, i.e. for v(i), $v \in V$, both v(i) and ¬v(i) hold.*

In **Figure 53**, the sources of the conflict are Dimensional analysis and Type checking and conversion, as mentioned.

Our implementation of conflict highlighting parses the UNSAT core using a recursive procedure starting at the root clauses of the core, traversing towards the logical sources of the conflict.

Intentions involved in the conflict are collected along the recursion. When a conflict occurs during backward analysis, our implementation highlights all intentions involved in the conflict as orange and intentions which are the logical sources of a conflict in red. An example can be seen in Figure 57, where the conflict previously described in Figure 53 is now shown through model highlighting. The highlighting is removed as the procedure backtracks to acquire more human judgments.

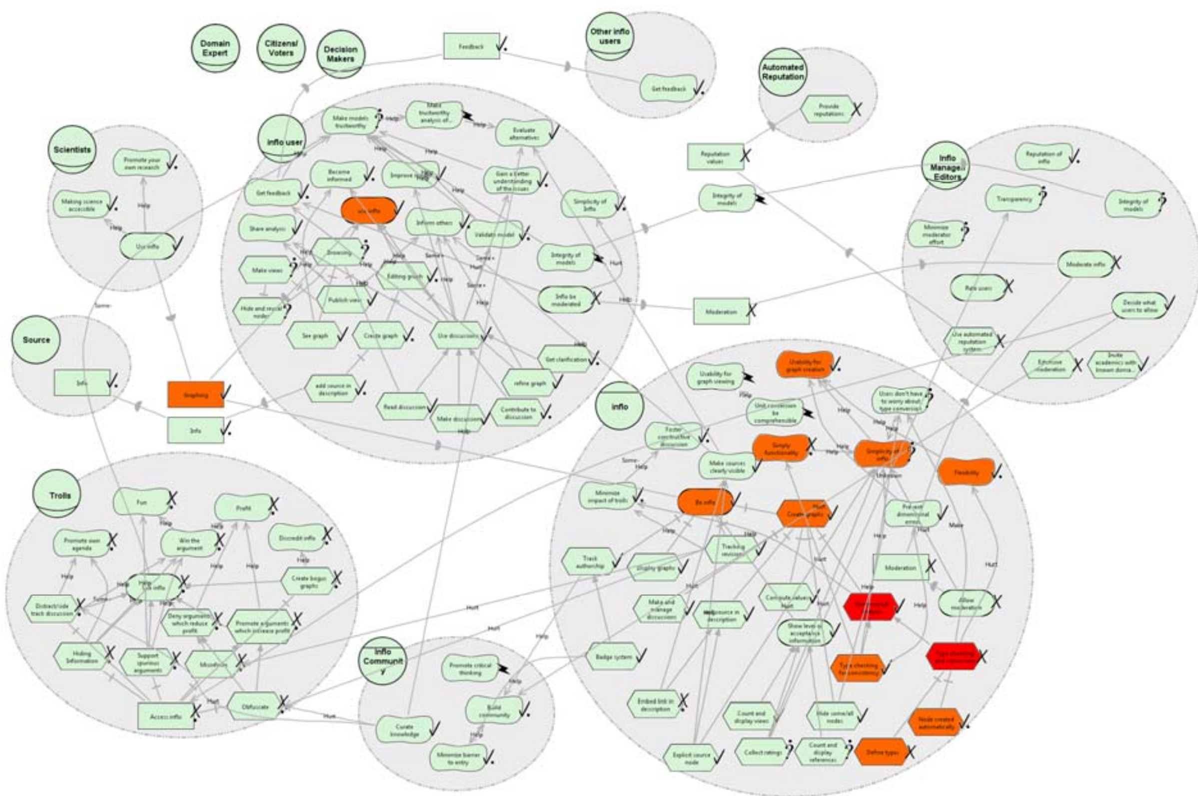


Figure 57 High-level view of Conflict Highlighting in the i* Model Resulting from the inflo Case Study

In addition to conflict highlighting, users are presented with a list of the intentions involved in the highlight along with the analysis value that these intentions would be assigned in order to produce the conflict. These values are extracted from the UNSAT core by storing the analysis value assigned to each predicate, with the logical sources having several conflicting values. Example output of this type corresponding to Figure 53 is shown in Figure 58.

The following intentions are involved in the conflict:

Simply functionality	PS
Type checking for consistency	PD, not PS
Flexibility	PS
use inflo	S, PS
Graphing	PS
Be inflo	PS
Create graphs	PS
Node created automatically	PD, not PS
Define types	PD, not PS
Usability for graph creation	PS
Simplicity of inflo	PS

The following intentions are the sources of the conflict:

Dimensional analysis	PS, not PS
Type checking and conversion	PD, not PS, PS

Figure 58 Clauses in the UNSAT core for the Conflict during Backward Analysis shown in Figure 53

9.3 Existing Goal Model Visualization Techniques

Any goal modeling framework which provides a visual syntax could be considered as a type of requirements visualization. However, work which focuses specifically on the visual aspects of goal modeling is especially relevant. Several techniques focus on using goal model visualizations in novel ways without making significant changes to the typical visual representation of goal graphs. For example, Sen and Jain use visualizations of activity cards and compilation tables to allow stakeholders to create and review goal decomposition hierarchies (Sen & Jain, 2007). Work in (Sawyer et al., 2007) uses i* models to capture various levels of concerns in dynamically adaptive systems without modification of i* visualization techniques. The approach of Rohleder (2008) provides a visualization technique to view the effects of Non-functional Requirements (NFRs) represented as goals on software variants.

Other work suggests modifications to the ways in which goal models are typically presented. For example, Ernst, Y. Yu, & Mylopoulos (2006) use various visualization interventions (e.g., line thickness, color, size) to visualize levels for software qualities such as trust and feasibility over goal models (Ernst, Y. Yu, & Mylopoulos, 2006). Results of quantitative goal analysis from (Giorgini et al., 2005) are displayed beside intentions, but are not otherwise used to affect visualization. Although the color intervention is similar to the visualizations used in our work, the meaning of the colors differs.

Further work focuses on improving i* syntax. Moody, Heymans, & Matulevicius, (2009) analyze i* syntax using principles for visual notation design such as semiotic clarity (one graphical symbol for each semantic concept). Although suggested changes to i* syntax may improve i* visualizations, our focus is on finding visualizations to support analysis, and not on improving the underlying visual representation of i*.

Unlike the mechanisms introduced in this chapter, these approaches do not focus on visualization of analysis results over i* models. The exception is the approach taken for GRL models in the jUCMNav tool, which uses varying shades of red and green to represent the level of satisfaction and denial resulting from automated qualitative and quantitative analysis (JUCMNav, 2011). These visualizations are potentially compatible with those introduced in the current work, assuming that new mechanisms for highlighting leaf /root and conflict intentions can be found (e.g., increasing size, thicker borders).

9.4 Conclusions

Although analysis over goal models can be helpful to better understand and improve models, increasing understanding of the domain, difficulties in model and analysis comprehension exist. We claim that these difficulties can be partially mitigated through visualization techniques. To our knowledge, no other work specifically addresses visualization aspects of goal model analysis.

We have described visualization techniques to address three specific analysis usability issues: finding starting points for analysis, finding intentions involved in judgments, and understanding model conflicts. Features allowing highlighting of model leaves and roots are provided. When a human judgment question is asked, intentions involved in the judgment are highlighted automatically. Mechanisms to highlight intentions involved in or acting as the logical source of conflicts are described, including relevant formal definitions of useful concepts.

The introduced visualizations are intended to support the comprehension of analysis results (R2), come with formal definitions (R7), aid in starting the iterative methodology (R12), and hide the procedure complexity through tool support (R16). These contributions are summarized in Figure 59. We test the claimed benefits of these techniques in a user studies described as part of Chapter 12.

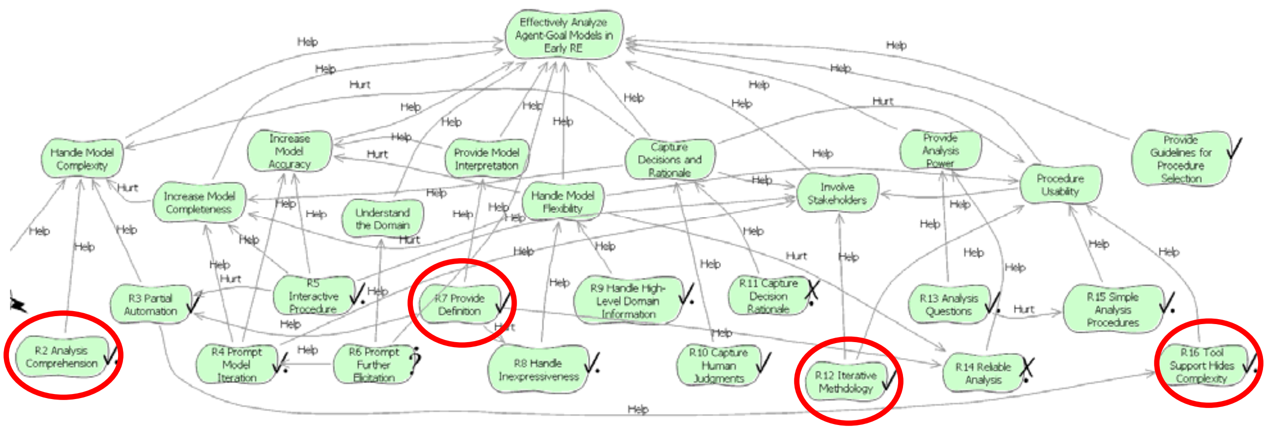


Figure 59 Focus of the Current Chapter in the Requirements for analysis of Agent-Goal Models in Early RE based on a Combination of Existing Work

Chapter 10 Detecting Judgment Inconsistencies to Encourage Model Iteration in Interactive i* Analysis

As we have stated in previous chapters, model analysis procedures which prompt stakeholder interaction and continuous model improvement are especially useful in early RE elicitation. Previous chapters have introduced qualitative, interactive forward and backward analysis procedures for i* models. In Chapter 12 we describe empirical studies which test the forward and backward analysis procedures against our goals for early RE goal model analysis. Studies with experienced modelers in complex domains have shown that this type of analysis prompts beneficial iterative revisions on the models. However, studies of novice modelers applying this type of analysis do not show a difference between semi-automatic analysis and ad-hoc analysis (not following any systematic procedure). In this chapter, we encode knowledge of the modeling syntax (modeling expertise) in the analysis procedure by performing consistency checks using the interactive judgments provided by users. We believe such checks will encourage beneficial model iteration as part of interactive analysis for both experienced and novice i* modelers.

10.1 Motivation

As stated in Chapters 1 and 2, because of the high-level, social nature of early RE models, it is important to provide procedures which prompt stakeholder involvement (interaction, R5) and model improvement (iteration, R4). To this end, previous chapters has introduced interactive, qualitative analysis procedures over agent-goal models (specifically, i* models) which aim to promote model iteration and convergent understanding. These procedures are interactive in that, where partial or conflicting analysis labels appear in the model, users are asked to provide a human input as resolution before the procedure proceeds further.

This chapter is an expansion of the following papers/reports:

Horkoff, J., & Yu, E. (2011c). Detecting Judgment Inconsistencies to Encourage Model Iteration in Interactive i Analysis. Proceedings of the 5th International i* Workshop (pp. 2-7).*

Experiences with skilled i* modelers in complex case studies have provided evidence that interactive analysis prompts further elicitation and beneficial model iteration. However, case studies comparing ad-hoc to semi-automated interactive analysis using novice participants showed that model iteration was not necessarily a consequence of systematic interactive analysis, but of careful examination of the model prompted by analysis in general. More details concerning these studies and their results can be found in Chapter 12. By analyzing study results, we conclude that the positive iterative effects of interactive analysis found in case studies are dependent upon modeling expertise (the ability to notice when analysis results were inconsistent with the model), domain expertise (the ability to notice when results differed from the modeler's understanding of the world), and interest in the domain being modeled (caring enough about the modeling process to improve the model).

One consequence of these results would be to recommend that interactive analysis be performed by, or in the presence of, someone with significant knowledge of i*. However, this is often not a reasonable expectation, as many i* modelers may be new to the notation and modeling technique, and will want to be guided by evaluation procedures in analyzing the model. As a result, we aim to embed some modeling expertise into the analysis procedure and corresponding tool support by detecting inconsistencies using the results of interactive judgments.

Case study experiences show that making judgments over the model can lead the modeler to revise the model when the decision made using domain knowledge differs from what is suggested by the model. For instance, in the simple example model used in Chapter 7, repeated in Figure 60, if the application Asks for Secret Question but does not Restrict Structure of Password, model analysis would suggest that Usability would be at least partially satisfied. If instead, the modeler thinks that Usability should be partially denied, this means the model is inaccurate or insufficient in some way. Perhaps, for example, Usability also requires hints about permitted password structure.

However, in our studies we found several occasions where novice modelers made judgments that were inconsistent with the structure of the model, and did not use these opportunities to make changes or additions to the model. To place this situation in the context of our previous example, if the Application Asks for Secret Question but does not Restrict Password the student may have decided that Usability was still partially denied, continuing the evaluation without modifying the model to be consistent with their judgment.

Similarly, our studies and experiences showed that it is easy to forget previous judgments over an intention element and to make new judgments which are inconsistent with previous judgments. For example, a user may decide that if Security is partially denied and Usability is partially satisfied, Attract Users is partially denied. In another round of analysis, if they are presented with an identical situation, they may now decide that Attract Users has a conflict.

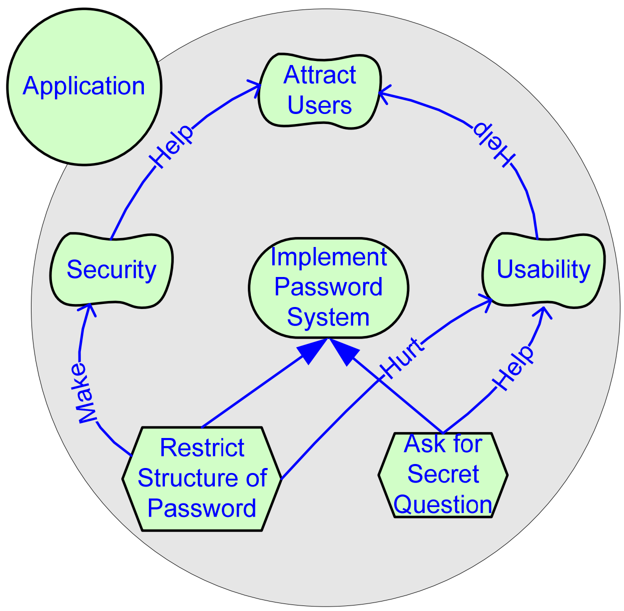


Figure 60 Simple Example of an i* Model for a Password System

We use these observations to guide us in embedding modeling expertise into interactive i* analysis by detecting inconsistencies using judgments. We distinguish and check for two types of inconsistencies: inconsistencies with the structure of the model and inconsistencies with judgment history. In this work, we take the initial steps of describing these checks formally and through examples. Future work will test the practical effectiveness of these checks in encouraging beneficial i* model iteration. We summarize the focus of this chapter in Figure 61.

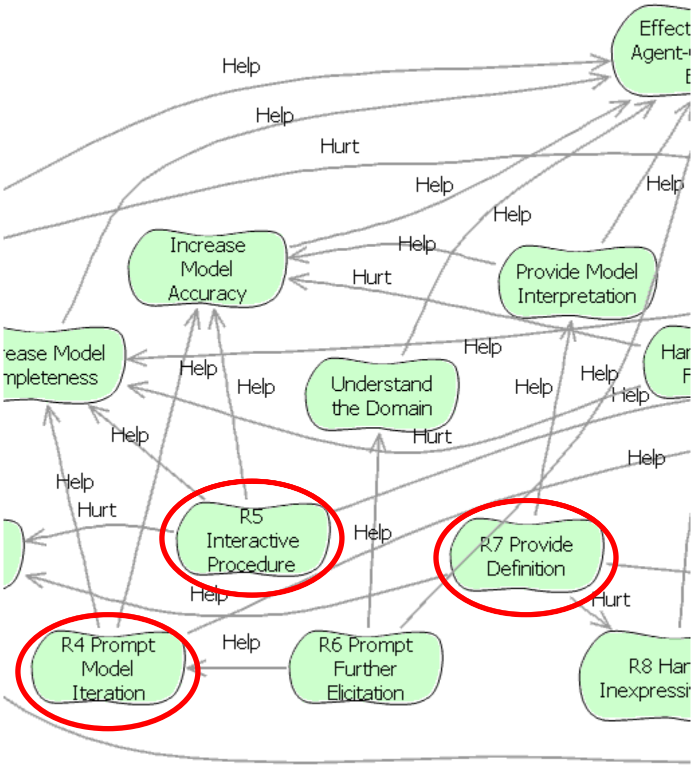


Figure 61 Focus of the Current Chapter in the Requirements for analysis of Agent-Goal Models in Early RE

10.2 Review: Propagation Axioms and Human Judgment

Recall that propagation can be described via the forward and backward propagation axioms described in Chapter 6 and Chapter 7. Generally, for an intention $i \in \mathcal{I}$, with is the destination of a relationship, $r \in \mathcal{R}$, $r: i_1 \times \ldots \times i_n \rightarrow i$ these predicates take on the form:

Forward Propagation:

(Some combination of $v(i_1)$ … $v(i_n)$, $v \in \mathcal{V}) \rightarrow v(i)$

Backward Propagation:

v(i) → *(Some combination of* $v(i_1)$ … $v(i_n)$, $v \in \mathcal{V})$

The interactive nature of the procedures comes when human judgment is needed to resolve incoming partial or conflicting labels (forward) or to provide feasible combinations of incoming labels to produce a target label (backward). New judgments are added to the model formalization by replacing the axioms defined above for an intention with new axioms of the same form, describing the judgment. For example, given *S(*Restrict Structure of Password*)* and *S(*Ask for Secret Question*)* (both alternatives are satisfied),we decide that Usability has a conflict, *C(*Usability*)*, we would remove all axioms having Usability as a target or source and add:

Forward: *S(*Restrict Structure of Password*)* ∧ *S(*Ask for Secret Question*)* → *C(*Usability*)*
Backward: *C(*Usability*)* → *S(*Restrict Structure of Password*)* ∧ *S(*Ask for Secret Question*)*

In this work we will refer to set of incoming or outgoing labels as the label bag (LB). This is the result of the left side of the forward propagation axioms as they are propagated through the links, and the set of resulting labels on the right side of the backward propagation axioms. We refer to the right of the forward axiom and left of the backward axiom as the individual label, *IL*. Forward judgments then consist of *LB* → *IL* and backward judgments consist of *IL* → *LB*.

10.3 Detecting Inconsistencies in Interactive Judgments

In this section we define two types of inconsistencies using human judgments.

10.3.1 Inconsistencies with Model

When considering inconsistencies between a judgment and the model, we compare the contents of the label bag (*LB*) to the individual label (*IL*), looking for inconsistencies. For example, if the label bag has no positive labels (*S*, *PS*) and the *IL* is positive, we classify this as inconsistent (Case 3). We enumerate the following cases which we define as inconsistent, summarizing each case in after the "//" symbols:

For a judgment LB → IL or IL → LB over $i \in \mathcal{I}$:

//there are no unknown labels in the LB, but the IL is unknown

Case 1: for all $v_j(i_j)$ in LB, $v_j \neq U$ and IL = U(i)

//there are no negative labels in the LB, but the IL is negative

Case 2: for all $v_j(i_j)$ in LB, ,$v_j \neq PD$ or D and IL = PD(i) or D(i)

//there are no positive labels in the LB, but the IL is positive

Case 3: for all $v_j(i_j)$ in LB, , $v_j \neq PS$ or S and IL = PS(i) or S(i)

//the LB is all positive or all negative, but the IL is a conflict

Case 4: for all $v_j(i_j)$ in LB, (v_j = PS or S) or (v_j = PD or D) and IL = C(i)

In the forward case, the label bag can be said to represent evidence from the model, while the individual label is the user judgment. In the backward case, the individual label is the required evidence in the model, while a permissible label bag is the user judgment applied to the model structure.

Example. We have already seen one example of this inconsistency type in the motivation section, if the modeler believes Usability should be partially denied, when the model suggests that Usability should be at least partially satisfied.

In another example, if both Security and Usability were somehow satisfied, that would suggest that Attract Users would be at least partially satisfied. If, when posed the corresponding human judgment question, the user instead said that Attract Users was unknown, this would be inconsistent with the propagation in the model. In fact, we can find example where users in our studies picked unknown as a resulting value when there were no incoming unknown values.

10.3.2 Inconsistencies with Judgment History

When considering inconsistencies with between old and new judgments over the same intentions, we compare the label bag (*LB*) in the new and previous judgments, looking for cases when the label bag is the same, is clearly more positive, or more negative, using the ordering of labels from (1). We use this comparison to decide whether the new individual label (*IL*) is consistent with the old individual label. An example of the case is described in Section 1, when the label bag is equal, but the individual label is not.

To aid in our definition of these cases we will refer to *ILnew* and *LBnew*, the most recent judgment for $i \in \mathcal{I}$, and *ILprev* and *LBprev* , the previous judgments for *i*. We define psuedocode to check for these types of inconsistencies as follows:

For a judgment LBnew → ILnew (backward: ILnew → LBnew) over $i \in \mathcal{I}$:

For each previous judgment LBprev → ILprev over $i \in \mathcal{I}$:

//compare labels in previous LBs to labels in new LB

For each $v_j(i_j) \in$ *LBprev,*

For $v_k(i_j) \in$ *LBnew, compare* $v_j(i_j)$ *to* $vk(i_j)$

Classify as: >, =, or <

LBnew → ILnew is inconsistent with LBprev → ILprev if:

//The new LB is more positive, but the IL is more negative

All classifications are > or =, and ILnew < ILprev

//The new LB is more negative, but the IL is more positive

All classifications are < or =, and ILnew > ILprev

//The new and old LBs are identical, but the IL has changed

All classification are = (LBprev = LBnew), and ILnew ≠ ILprev

Example. In an example, the user decides that with incoming labels of *PS(*Security*)* and *PD(*Usability*)*, Attract Users is *C(*Attract Users*)*. In the next round of evaluation, incoming labels may be PS(Security) and *C(*Usability*)*. If the user decided Attract Users was partially denied this would be inconsistent with the previous judgment. The new label bag is more positive than the previous, as *C* > *PD*, so the individual label should not be less than the previous individual label, *C*, i.e. not *U*, *PD*, or *D*.

10.3.3 Related Work: Inconsistencies and Conflicts in RE

Existing work has looked at the presence of inconsistencies or conflicts gathered as part of a requirements process. In Section 2.3.1.3, we have summarized work by (van Lamsweerde, Darimont, & Letier, 1998), which has identified various types of inconsistencies specific to artifacts expressed in KAOS (e.g., boundary conditions, ordering). This work outlines how conflict detection and resolution can be make part of the goal oriented RE process and suggests possible ways to mitigate or resolve the conflict. A related approach has been summarized in the same section, providing an overview of work falling under the category of Requirements Interaction Management (Robinson, Pawlowski, & Volkov, 2003).

Although the detection and mitigation of conflicts between requirement and goals is relevant to the approach introduced in this chapter, the focus of the consistency checks introduced in the previous section is not the detection of inconsistencies between goals. Contribution links in the i* framework, already allow for the direct or indirect representation of conflicts or tradeoffs between intentions. See Figure 62 for samples of how conflicts can be represented in i*. On the left, a direct conflict is shown between R13 and R15 (note this is actually a hurt link in our Early RE Requirements Goal Model). On the right, an indirect conflict is shown between Provide Model Interpretation and Accommodate Model Flexibility in regards to Increase Model Accuracy. The latter conflict is not a conflict in the logical sense (it is impossible to satisfy both intentions at once) but is instead a partial or qualitative indication of a tradeoff between requirements.

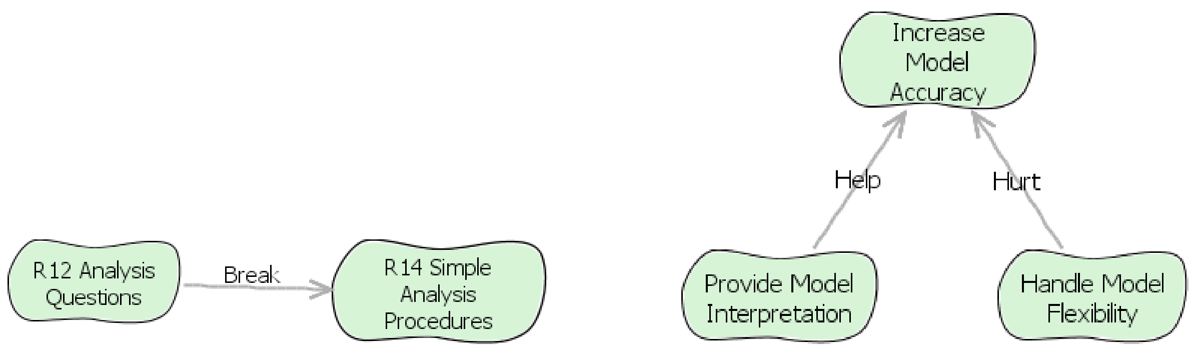


Figure 62 Examples of a Conflict between Goals in i* as Represented by Contribution Links

Contribution links are represented in the analysis procedure via propagation axioms, as described in Chapter 6 and Chapter 7. The analysis predicates as defined in this work do not, in isolation, produce model representations which are inconsistent, i.e. over which a solution cannot be found. This is because all possible predicates representing analysis values can be true over the same intention at once (*PS(i), S(i), PD(i), U(i),* etc., $i \in \mathcal{I}$). Logical inconsistencies in the model can arise when target values are added to the model. For example, an intention has a target of *PS(i)*, but all possible solutions also produce ¬*PS(i)*. This inconsistency would be found and reported to the user as part of the backward algorithm (which also includes forward propagation).

When encoding human judgment, the judgments are added to the model by adding new axioms (removing previous axioms describing propagation over the same area). It is possible for an axiom representing a human judgment, in conjunction with a target value, to produce a logical inconsistency in the model. However, this inconsistency is found as part of backward analysis (Chapter 7) and is not the focus of this chapter. This chapter instead focuses on checking for consistency between and amongst analysis label judgments over goals and the model, and not between goals themselves. We examine inconsistencies between the user's judgment over the domain and the analysis results over a model. This judgment may or may not produce a logical inconsistency with a target in the model.

This difference between inter-goal conflicts and conflicts between judgments and analysis results means that approaches for classifying, detecting, and mitigating inconsistencies introduced in previous work may not be directly applicable to the types of inconsistencies detected in this chapter. In fact, we have formally described detection of selected inconsistencies in the previous section. We discuss the applicability of inconsistency classification and mitigation approaches in the next section.

10.3.4 Discussion

Choice of Consistency Checks. We have been very flexible and permissive in defining our judgments, only defining cases which are clearly inconsistent. For example, we could include rules to measure when a CL is mostly negative (many more negative labels than positive), and check that the IL is at least partially negative. In this case, we would need to set specific quantitative measures to compare the negative and positive evidence (e.g., if three more positive labels than negative, 50% more negative than positive). Such measures are fairly arbitrary. It is desirable to avoid providing too many warnings over judgment, less the users begin to ignore them. Future studies can test the optimal level of strictness over judgment checks by running more user studies, similar to what is done in Chapter 12.

Inconsistency Classification (Reasons for Inconsistencies). As we have alluded to in the motivation section, inconsistencies between the model and judgments, or amongst judgments could arise for a variety of reasons. The empirical studies described in Chapter 12 provided evidence to show that inexperienced modelers made inconsistent judgments. Although the participants of individual case studies followed a think-aloud protocol, we did not consistently directly ask the participants about the reasons for their decisions. Therefore, we can only use our knowledge of the procedure and experiences in user studies to speculate over the underlying reasons for model/judgment and judgment/judgment inconsistencies. Future work user studies should ask more probing questions in inconsistent judgment situations.

Reasons for inconsistency between judgments may include:

- The user has changed their mind from one judgment to the next.
- The user has misunderstood the question in one instance (e.g., did not notice a link type, did not consider a label).
- A different user(s) makes the judgment.
- Discussions as part of the interactive iterative analysis may have resulted in the user learning new information about the domain, effecting their decision.

Reasons for inconsistency between the model and judgments can include:

- **Judgment and model conflict.** The user's domain knowledge produces a decision which conflicts with the structure of the model as represented by the analysis results. We have provided examples of this scenario in Section 10.3.1. This situation could lead to two potential results:
    - The users would use this inconsistency as motivation to iterate over and make positive changes to the model.
    - The users conclude that the model concepts are not sufficiently expressive enough to reflect the reasons behind their judgment. In this case we would recommend making note of the deficiency, potentially using other modeling or textual RE artifacts (tables, templates) to capture this information.
- **Judgment and analysis results conflict.** The user's domain knowledge produces a decision which conflicts with the analysis results, i.e. the user thinks the model is right, but the analysis results are wrong. This situation could lead to two potential results:
    - The users decide the analysis procedure is incorrect in some way. Collecting examples of this case would allow improvement of the interactive procedures in future work.
    - The users conclude that the analysis results are not sufficiently expressive enough to reflect the reasons behind their judgment. In this case we make similar

recommendation to the earlier case, make a note of the deficiencies and then use another technique which covers this information. For example, if the analysis cannot express a certain condition necessary for a judgment, this condition could be added to a later (more focused) representation of the model in a more expressive notation, such as KAOS for formal Tropos.

- **Varying interpretation of domain concepts.** It is possible that intended meaning of certain intentions is not the meaning understood by the person(s) making the judgment. If, for example, a different person(s) constructed the (part of the) model than the person(s) making the judgment. Terminological interference in goal models is explored further in (Niu & Easterbrook, 2006) (in fact this work uses the same counseling service case study as used in this work). The reparatory grid technique introduced in this work could be used to analyze and potentially resolve terminological inconsistencies. This could also produce useful changes to the model.
- **Random selection.** The user "gives up", does not understand the question, or does not know the answer, and, as such, randomly picks a label. Several examples of this situation were found in the individual case studies. We reason that this situation is due in part to usability issues in the procedure, but also to the artificial nature of the case studies. We discuss this further in Chapter 12.

Approaches which examine requirement inconsistency in RE (discussed in the previous section) have introduced several types of inconsistencies including a process-level deviation, a terminology clash, a logical inconsistency, negative interactions, and implementation conflicts. The inexpressiveness of i* when compared to KAOS (e.g., lack of temporal information) prevents us from applying most of these classifications directly to the inconsistencies detected in this chapter. However, some inspiration can still be gained from these categories. As described, terminological inconsistency is a potential cause of judgment inconsistencies in agent-goal models. In considering logical inconsistency, we have specified that the inconsistency we are searching for in this chapter is not a logical consistency (this is covered in the backward procedure) but instead an inconsistency between the users judgment and the model or analysis results.

Resolving Inconsistencies. Although we have defined inconsistent judgment situations, we have not specified what actions to take when inconsistencies are found. In order to provide flexibility, we do not recommend preventing users from making inconsistent judgments, but instead suggest warning users, either when the judgment is made, or after the fact using a judgment consistency checker. This feature would work similarly to a built-in model syntax checker. Both the judgment consistency and model syntax checks are currently being implemented in the OpenOME tool. The GMF meta-model of the tool has been expanded to include judgment and evaluation alternatives. More detail concerning the metamodel and implementation will be provided in Chapter 11.

Existing work (van Lamsweerde et al., 1998; Robinson et al., 2003) provides suggestions for conflict mitigation between goals, including avoiding boundary conditions, weakening a goal, and finding alternative goal refinements, goal relaxation, compromise, or restructuring. Although most of these mitigations are applicable for resolving inter-goal conflicts in early RE agent-goal models, they are less applicable to conflicts between the model and a judgment. In this case, the mitigations would be less about finding new ways in which targets can be satisfied more about making the model more consistent with domain understanding. The purpose is to improve the accuracy of the model, and not to improve its satisfiability. Future work should design further user studies to examine whether there are patterns or classifications amongst the types of changes made to make the model and judgments more consistent.

10.3.5 Conclusions

This work reinforces the semantics of i* by embedding rules into the iterative analysis procedures which check for consistency amongst and between user judgments in the model. The aim is to encode some modeling expertise into the tool which implements these checks. In this way, we intend to further increase user involvement in the analysis process, as they potentially revise the model or their judgments (interaction, R5) and to further encourage model improvement as a result of these changes (iteration, R4). A precise definition of these checks has been provided (R7). We summarize these contributions in Figure 63.

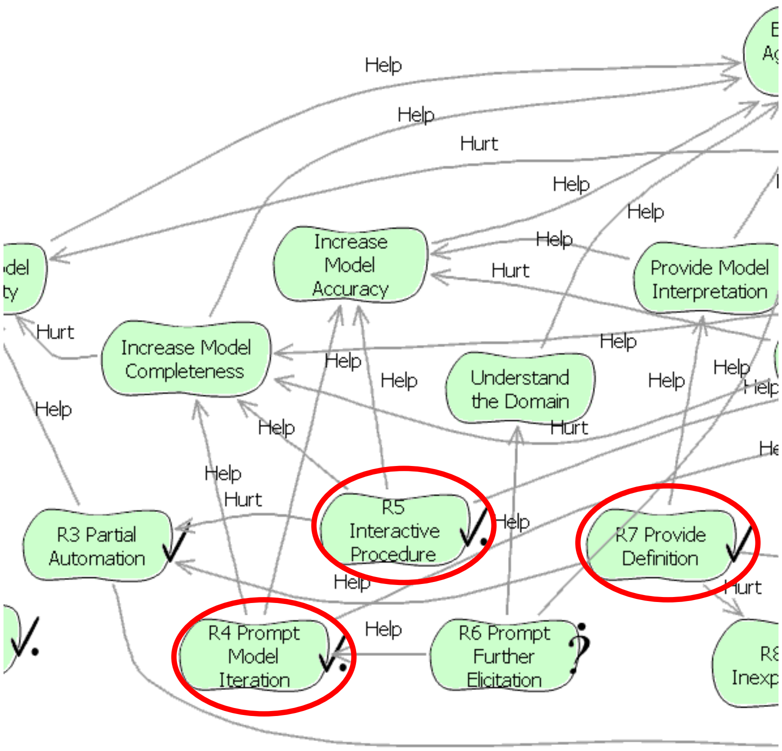


Figure 63 Focus of the Current Chapter in the Requirements for analysis of Agent-Goal Models in Early RE

Chapter 11 Tool Support for Iterative, Interactive Agent-Goal Model Analysis

The analysis framework has been implemented in OpenOME, an open-source requirements modeling tool. The tool is and Eclipse-based application, making use of the Eclipse and Graphical Modeling Frameworks (EMF and GMF). Previous work allowed the tool to support i* modeling, but expansions have been made in the tool to support the framework described in this thesis. Specifically the following features have been implemented:

- Easier access to qualitative evaluation labels
- Interactive forward analysis over i* models
- Interactive backward analysis over i* models
- Storage of analysis results
- Storage and management of human judgment results
- Visualization interventions for comprehension of analysis results
- General analysis usability (quick access to labels, procedures, highlighting)
- Syntax checking
- Importing/Exporting to q7 and iStarML formats
- Consistency checks over human judgment
- Tabular view of i* models

This chapter describes summarizes the implementation of OpenOME features in more detail, including the implementation metamodel. OpenOME was originally ported from the OME tool

This chapter is an expansion of the following papers/reports:

Horkoff, J., Yu, Y., & Yu, E. (2011). OpenOME: An Open-source Goal and Agent-Oriented Model Drawing and Analysis Tool. iStar Tool Fair: Proceedings of the 5th International i Workshop (pp. 154-156).*

to the Eclipse platform by Yijun Yu. Several features were added by Yijun and Neil Ernst as part of various projects. More features have been implemented by undergraduate students as part of a project course or undergraduate project, supervised by the thesis author. A list of such undergraduate students working on OpenOME is included in Table 73 in Appendix D. A list of past individual contributors is also provided on the Trac Wiki: https://se.cs.toronto.edu/trac/ome/wiki/Contributors.

Work described in this chapter address several of our requirements for early RE agent-goal model analysis. Most prominently, tool support hides complexity (R16) and provides partial automation (R3). The implementation aims to support large models, addressing the scalability of the analysis procedures (R1). Indirectly, the implementation facilitates the satisfaction of other goals, for example, it allows the user to capture and manage human judgments. We attribute the achievement of such goals to the chapter which introduces the conceptual mechanisms (the concept and formal description of human judgment) instead of to the implementation which allows its use in practice.

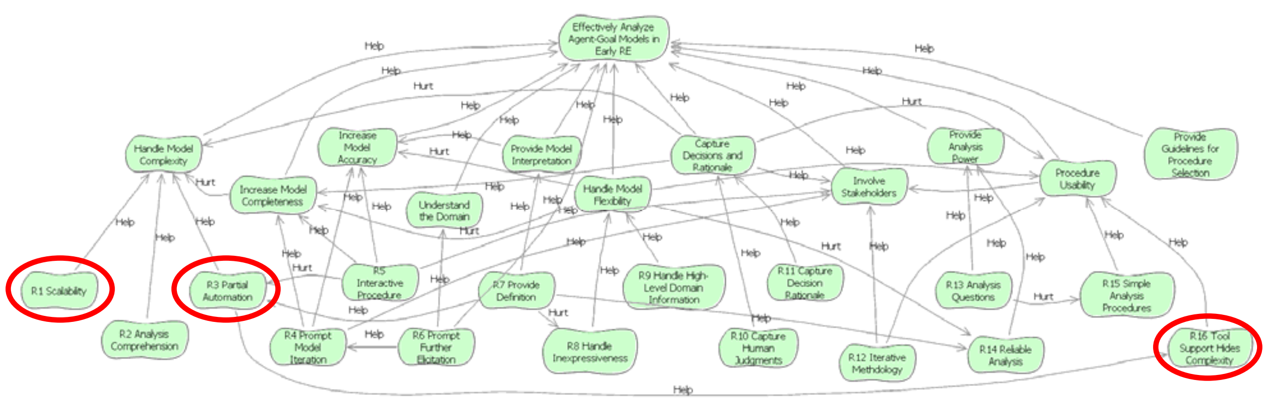


Figure 64 Focus of the Current Chapter in the Requirements for analysis of Agent-Goal Models in Early RE

11.1 OpenOME Background

OpenOME is a tool for the creation and analysis of goal and agent-oriented models as part of a systems analysis process. The tool supports modeling of the social and intentional viewpoint of a system, allowing users to capture the motivations behind system development in a graphical form. Creation and analysis of agent-goal models supports requirements elicitation, exploration, communication and trade-off analysis. OpenOME is especially useful to support elicitation and analysis in early requirement phases, where important non-functional and social information is made explicit.

OpenOME development originated in OME, a desktop Java application developed at the University of Toronto to support i* modeling ("OME, Organization Modelling Environment," 2008). In 2004, development of OME ceased and the source code was ported to the Eclipse platform, creating an open source version taking advantage of the Eclipse Modeling and Graphical Modeling Frameworks (EMF & GMF). Use of these frameworks allows us to automatically generate model editing code from an i* metamodel. This code has been customized and expanded to support features specific to i* modeling (e.g., opening and collapsing of actors). All code is written in Java. OpenOME architecture takes advantage of the Eclipse package development, allowing for extension or customization with the addition of a new development package. After several rounds of development, the current architecture no longer bears similarity to that of the OME tool. Current development is at version 3.4.1.

Windows, Linux and Mac releases of OpenOME can be downloaded from Sourceforge ("OpenOME, Sourceforge," 2011). User documentation and tutorials are available under the User Links section here on the OpenOME development Trac Wiki page ("OpenOME Trac Wiki," 2011). After several rounds of user studies, and through use of the tool in several systems analysis courses, OpenOME has reached a relatively mature and stable state. Bug reports and suggestions for new features can be sent to openome-support@cs.toronto.edu. More information on OpenOME details not covered in this chapter can be found on the developer wiki ("OpenOME Trac Wiki," 2011).

11.2 OpenOME Features

In the following section, we provide a brief description of each of the prominent features of OpenOME, with a focus on those directly facilitating interactive analysis. We provide more detailed breakdown of the interface showing the location of view and tabs corresponding to certain features in Figure 65.

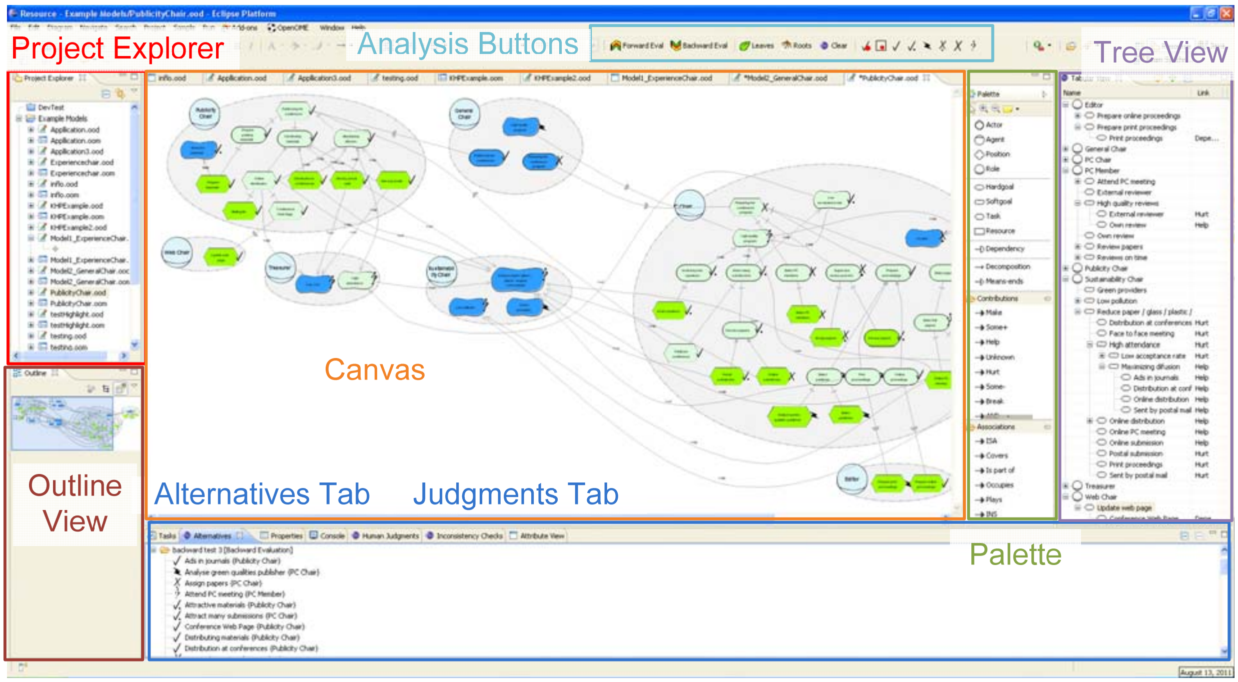


Figure 65 Screenshot of the OpenOME Tool Indentifying Feature Layout

11.2.1 i* Modeling

The tool allows users to draw models using the i* Framework syntax described in Chapter 5. Users can create i* models graphically on a canvas using a palette of shapes, see Figure 65 for the canvas and palette locations. Standard features such as saving, zoom, cut, copy, and paste are provided. Standard Eclipse features provide an outline view of the model. Models are grouped under user-created projects, shown in a folder view (Project Explorer).

Customized features specific to i* have been added to the tool: expanding and collapsing actors (from SD to SR form), clicking and dragging intentions inside and outside an actor, changing existing intention or link type, and straightening links,

11.2.2 Forward Analysis

OpenOME supports the forward and backward interactive, qualitative i* analysis procedures described in Chapter 6. Users can assign qualitative labels to intentions which represent their initial analysis question and then propagate these labels in a forward (direction of the link) or backward direction. When user input is needed for human judgment a pop-up window appears collecting the judgment from the user. A view of OpenOME during a forward judgment over a conference planning model is provided in Figure 66. Figure 67 shows a zoomed in view of the pop-up which asks the user for judgment. In this example Financial Success in the General Chair actor has two incoming values, Partially Satisfied from Involvement in Industry and Partially Denied from Sustainability. The user can pick one of any of the analysis labels from the list provided.

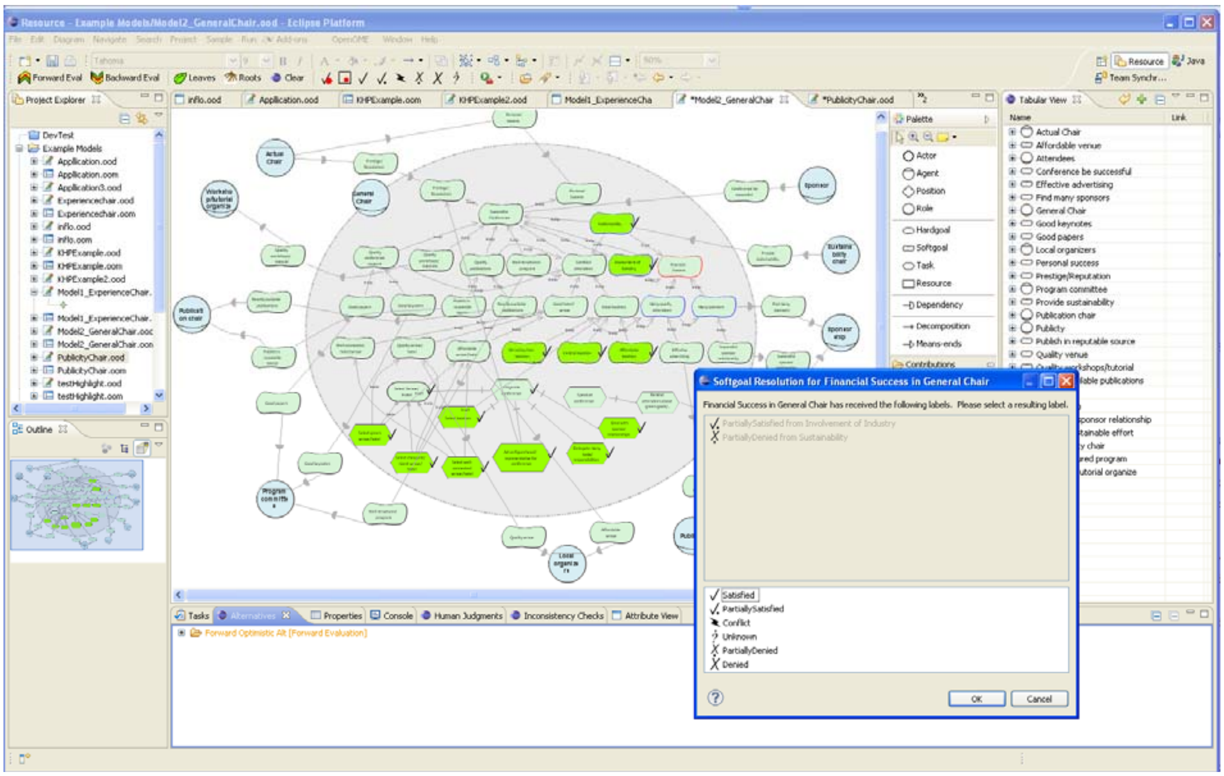


Figure 66 Screenshot of Forward Human Judgment in OpenOME

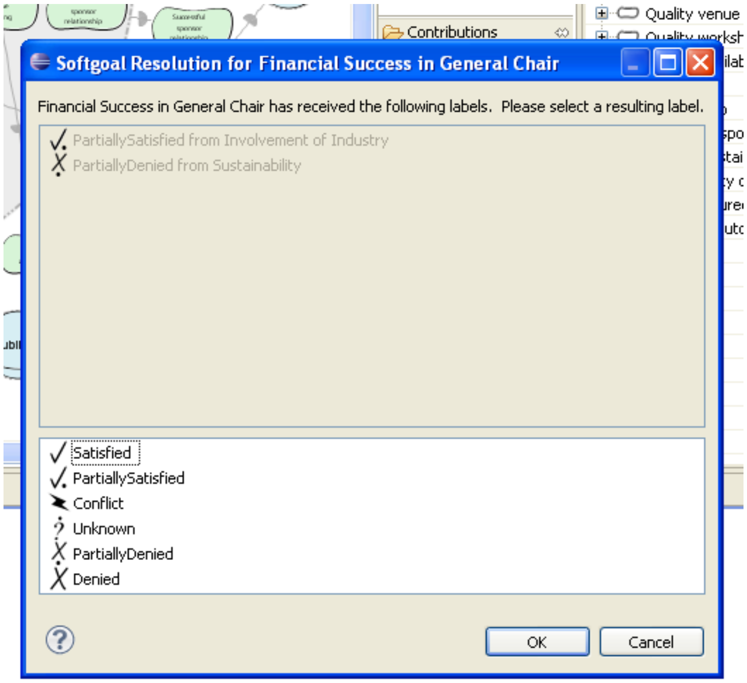


Figure 67 Screenshot of Forward Human Judgment Question Pop-up Window in OpenOME

Psuedocode of the forward analysis algorithm has been provided in Chapter 6. Analysis code is contained with the edu.toronto.cs.openome.evaluation package within the OpenOME code base. A detailed description of each implementation class goes beyond the detail needed for this thesis. The code itself contains useful comments describing the implementation.

11.2.3 Backward Analysis

OpenOME implements backward analysis as described in Chapter 7. Users can place targets on the model using available analysis labels, potentially aided by automatic root highlighting. Backward analysis works iteratively, calling the SAT solver, looking for intentions needing human judgment, collecting judgment with a pop-up window, and then re-running the SAT solver with human judgment encoded. See Figure 68 for a screenshot of backward analysis showing a human judgment pop-up and Figure 69 for closer view of the human judgment pop-up.

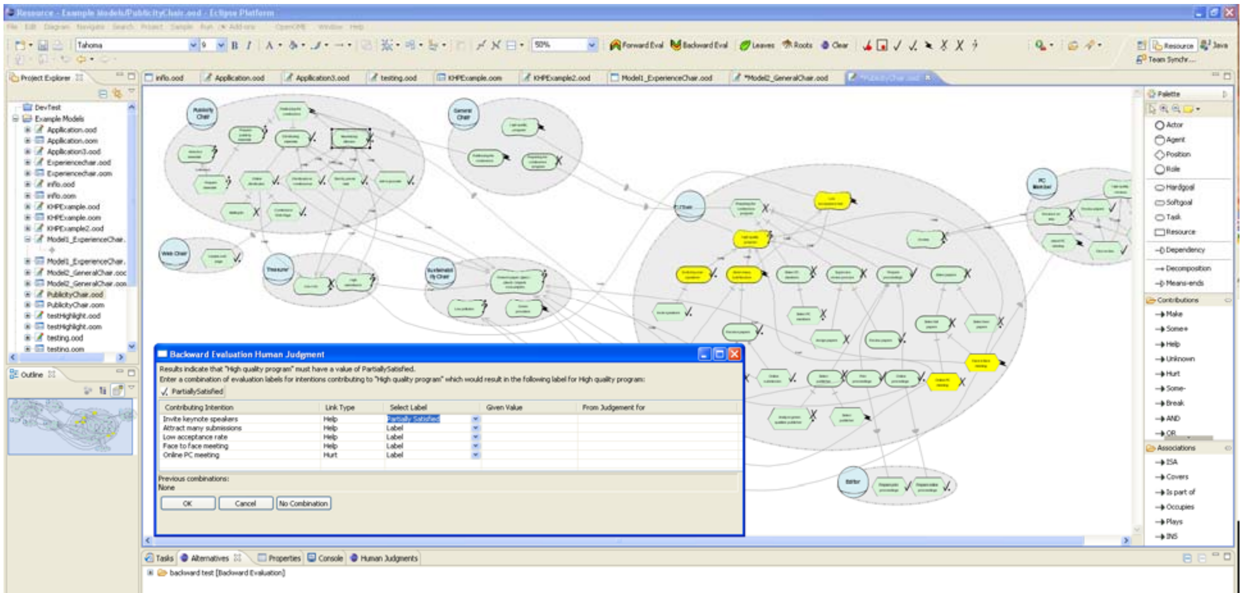


Figure 68 Example Backward Analysis in OpenOME

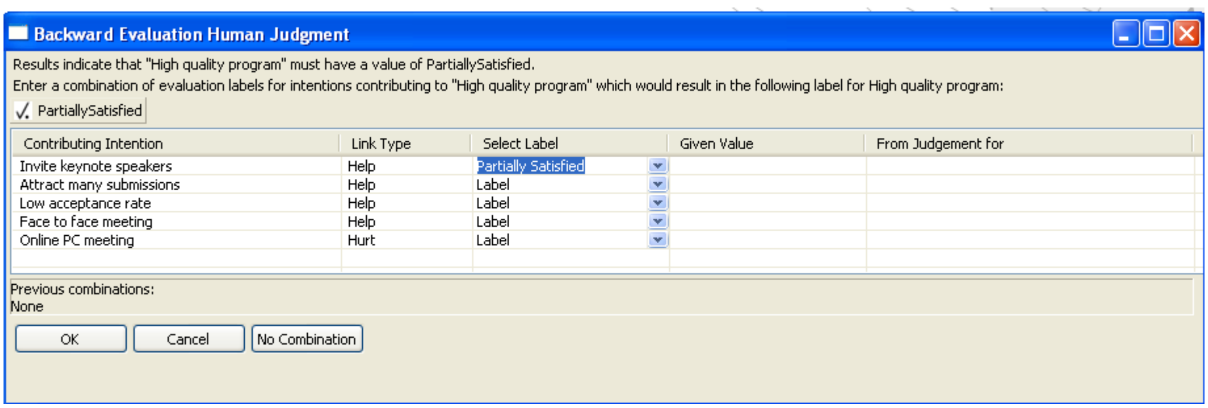


Figure 69 Example Backward Analysis Human Judgment Pop-up in OpenOME

The backward human judgment window will show previous judgments made over the same intention using text at the bottom of the pop-up window. We can see an example over the Application example used previous in Figure 70. Here two previous judgments have been given for the security softgoal, both resulting in conflicts (no results), listed at the bottom of the pop-up window (Previous combinations). With this feature, we are reminding users of their past judgments, helping them to not repeat themselves. However, we do not enforce that old judgments cannot be made again, as the user can backtrack across judgments over several alternatives, we let the user decide what decisions to change. Future improvements could allow them to reuse old judgments faster by selecting them from the list shown below.

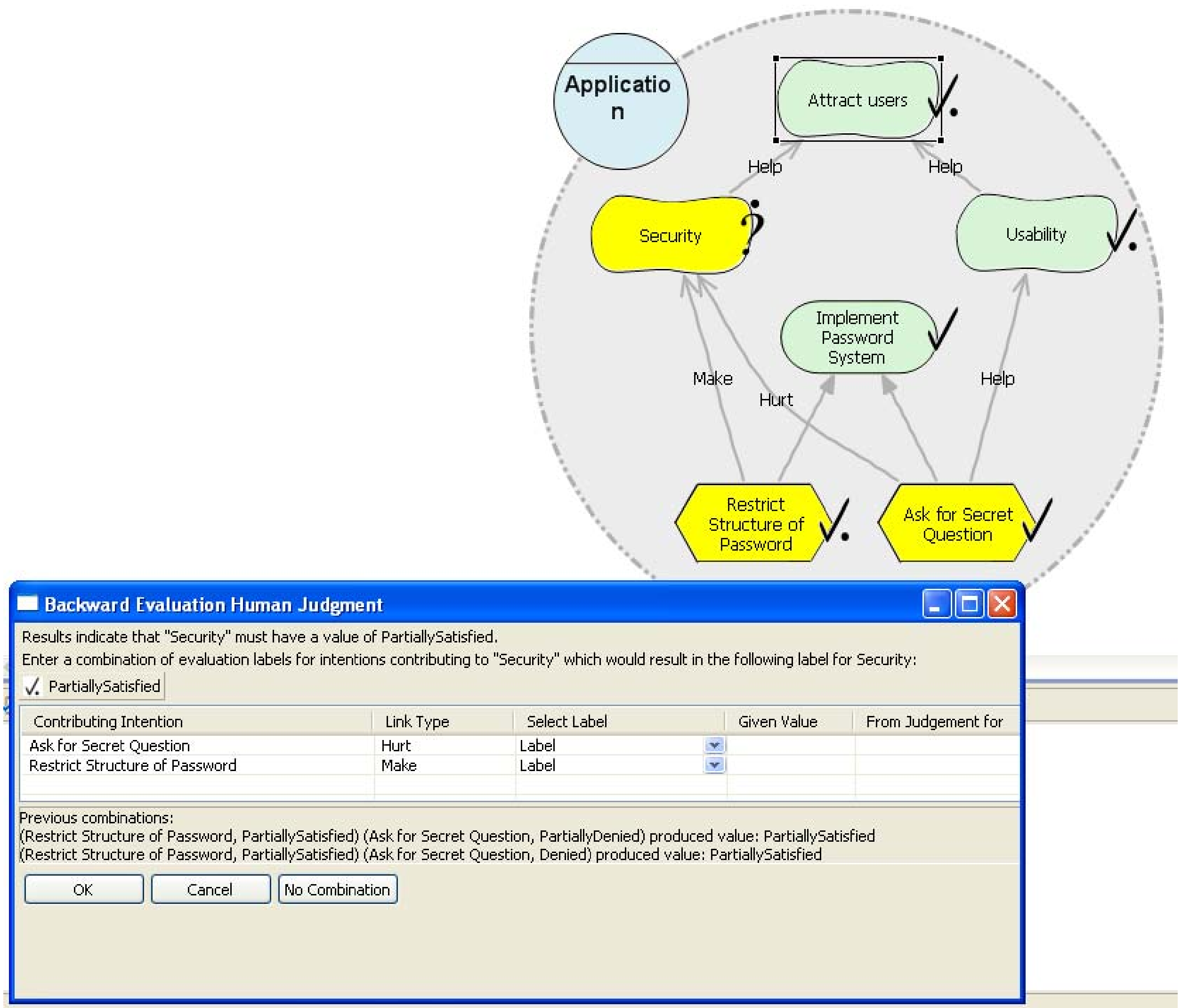


Figure 70 Backward Analysis Human Judgment Pop-up Window Example showing Previous Judgments for an Intention

The backward judgment window also shows users analysis labels given to the same intention as part of another judgment in that round of analysis. These values are removed as the procedure backtracks over the other judgments. For example, in Figure 71, we see a judgment window for Many Quality Attendees. In this case both Good hotel/venue and Good location have already been assigned Partially Satisfied in the judgment for Satisfied Attendees. The user does not have to give these two intentions partially satisfied labels in this judgment, but if they do not a conflict is likely to occur. If, for example, Good location is given partially satisfied in one judgment and partially denied in another, the new human judgment axioms and constraints will produce a conflict.

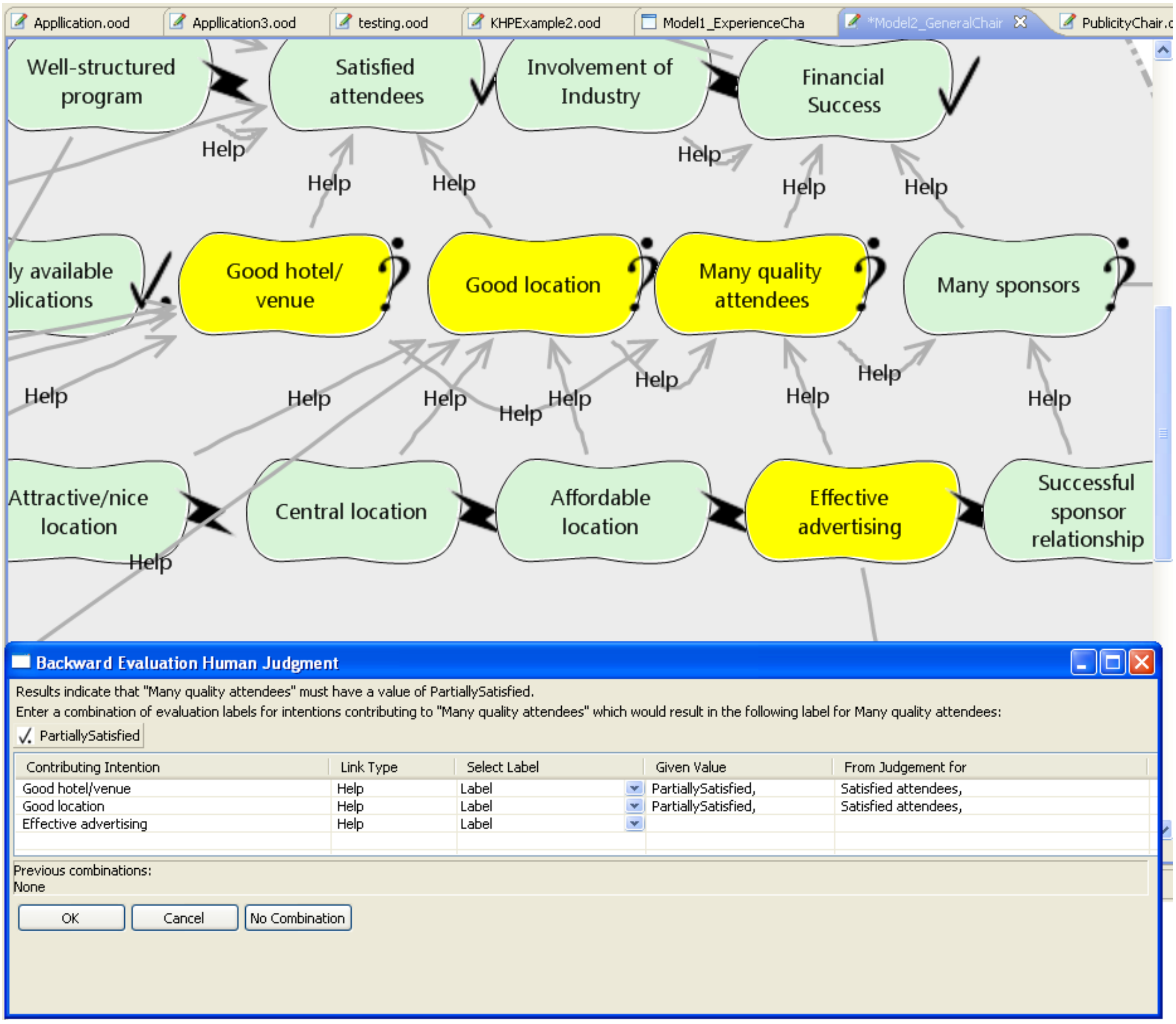


Figure 71 Backward Analysis Human Judgment Pop-up Window First Example showing Previous value for Intentions via other Judgments

In a second example, in a judgment situation for Financial Success, Sustainability has already been given a partially satisfied label in a judgment for Satisfied Attendees (Figure 72). In this case, as Sustainability hurts Financial Success, the user is put in a difficult position, they must relax their constraints for Financial Success, compensating with the other contributions, or make a judgment which produces a conflict.

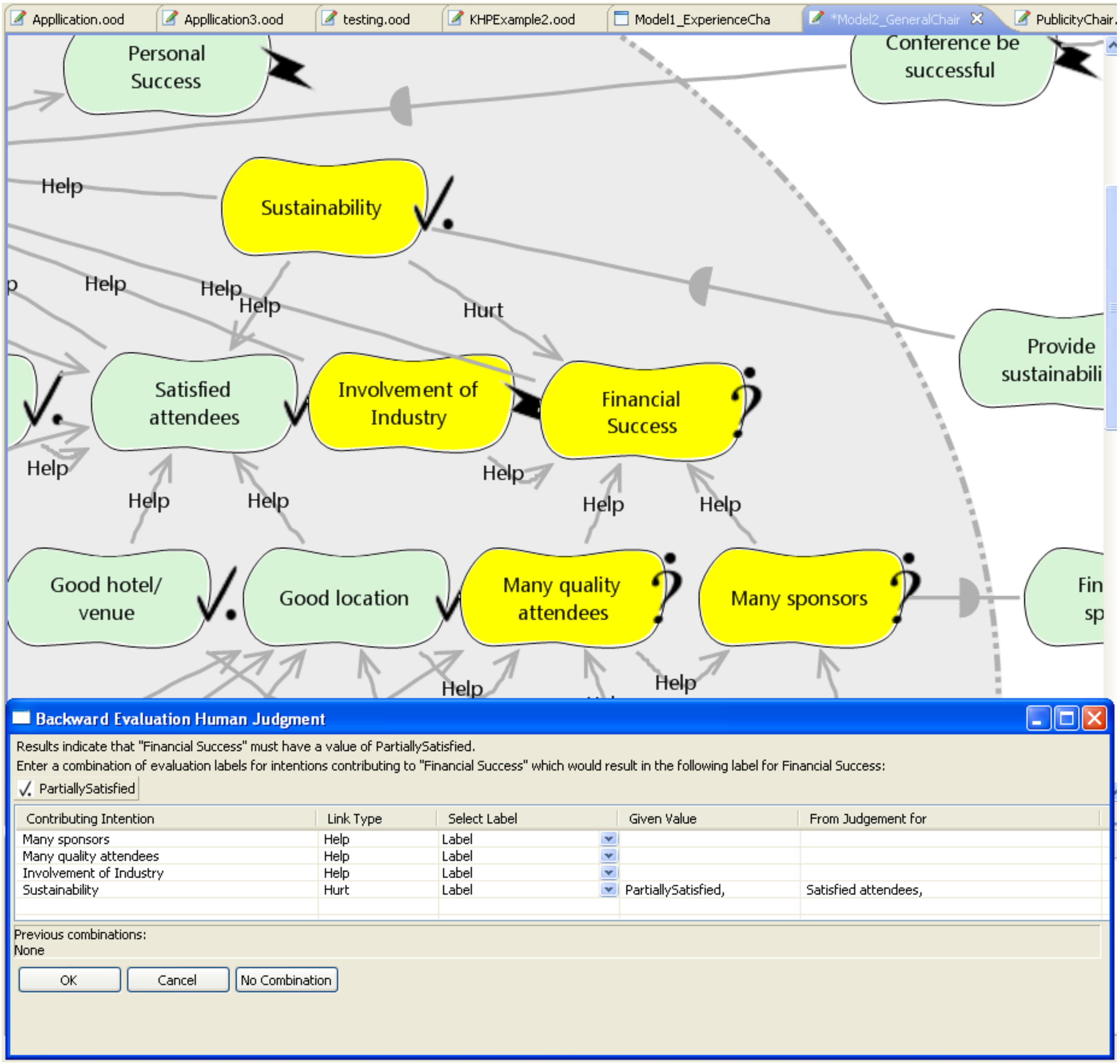


Figure 72 Backward Analysis Human Judgment Pop-up Window Second Example showing Previous value for Intentions via other Judgments

If the SAT solver cannot find a solution, minSAT is ran in order to find the UNSAT core. This core is displayed to the user via another pop-up window and intention highlighting. See Figure 73 for a screenshot of OpenOME showing a backward analysis conflict and pop-up window, and Figure 74 for a closer view of the conflict pop-up.

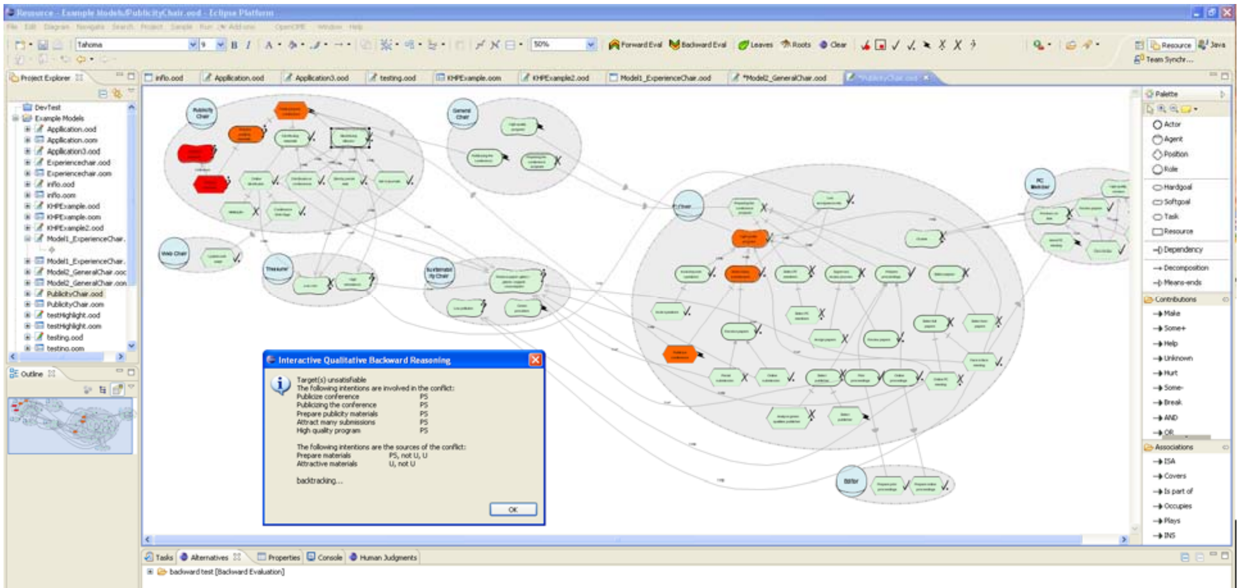


Figure 73 Example Backward Analysis Conflict in OpenOME

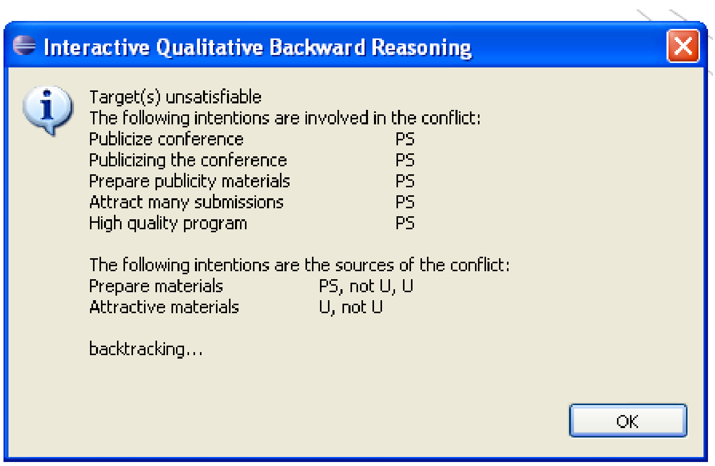


Figure 74 Example Backward Analysis Conflict Pop-up in OpenOME

More detail concerning the backward analysis algorithm can be found in Section 7.4.6.

11.2.4 General Analysis Improvements

One of the student development projects over OpenOME involved general improvements to analysis functionality, based on results gained from user studies described in Chapter 12. These improvements included buttons for quick addition and removal of analysis labels (previously available via context menu); buttons for forward and backward analysis; buttons for highlighting leaves and roots and removing highlights; improvements to the human judgment pop-ups, including adding icons for the analysis labels; and the underlying changing required for intention highlighting. Many of these improvements can be seen in the screenshot of the OpenOME toolbar repeated from Figure 54 below in Figure 75.

Figure 75 Screenshot of OpenOME Toolbar Showing quick Access to Leaf and Root Highlighting

11.2.5 Alternatives View

Results from multiple evaluations, including judgments, are stored in the metamodel, and can be viewed via the Alternatives Tab. Figure 65 shows the alternatives tab on the bottom of the figure, highlighted with a blue box. A more detailed example is shown in Figure 76. Each analysis alternative is listed in a tree view, showing the alternative name and the type of analysis (forward/backward). Trees can be expanded to see each intention in the model. Intentions with human judgment can be expanded to see a list of the judgments for that intention for that alternative. Double clicking on the judgment will open it. New improvements to the tool allow users to change the judgment. However, the tool will not yet re-prompt interactive analysis when judgments are changed. Handling model changes in the analysis framework is left to future work.

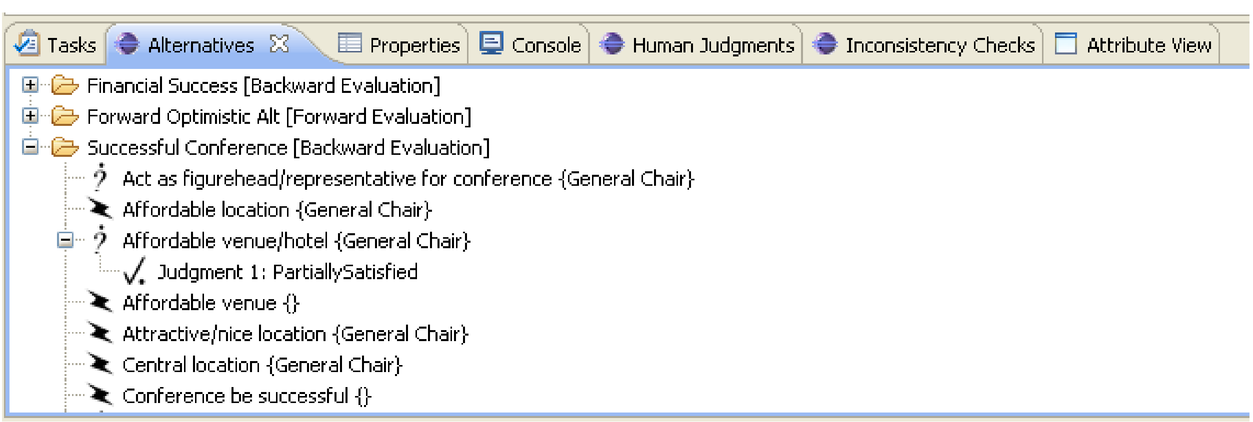


Figure 76 Detailed Example of the Alternatives Tab

11.2.6 Human Judgment Management

As judgments are made over the model, they are stored in each diagram file automatically, after having been added to the OpenOME metamodel. Each judgment belongs to an intention, has a results, and a label bag containing incoming labels and their destinations. Stored previous judgments are reused automatically in the forward procedure across one alternative analysis. Judgments are not automatically reapplied in backward analysis. Future improvements should show users all past judgments, and allow them to optionally apply them in user-friendly way.

Previous judgments for each intention can already be viewed under each analysis alternative in the Alternatives View. However, there is a need to provide for a more global view of all judgments across alternatives, to allow for comparison and consistency checks, as described in Chapter 10. As such, a Human Judgment Tab has been added to the tool, listing all intentions which have been given judgments in a tree view. Each intention can be expanded to view the judgments given. Each listed judgment shows the result and from which alternative analysis it was derived. Double clicking on a judgment opens the judgment for viewing. We can see a high-level and detailed view of the Human Judgment Tab in Figure 77 and Figure 78, respectively.

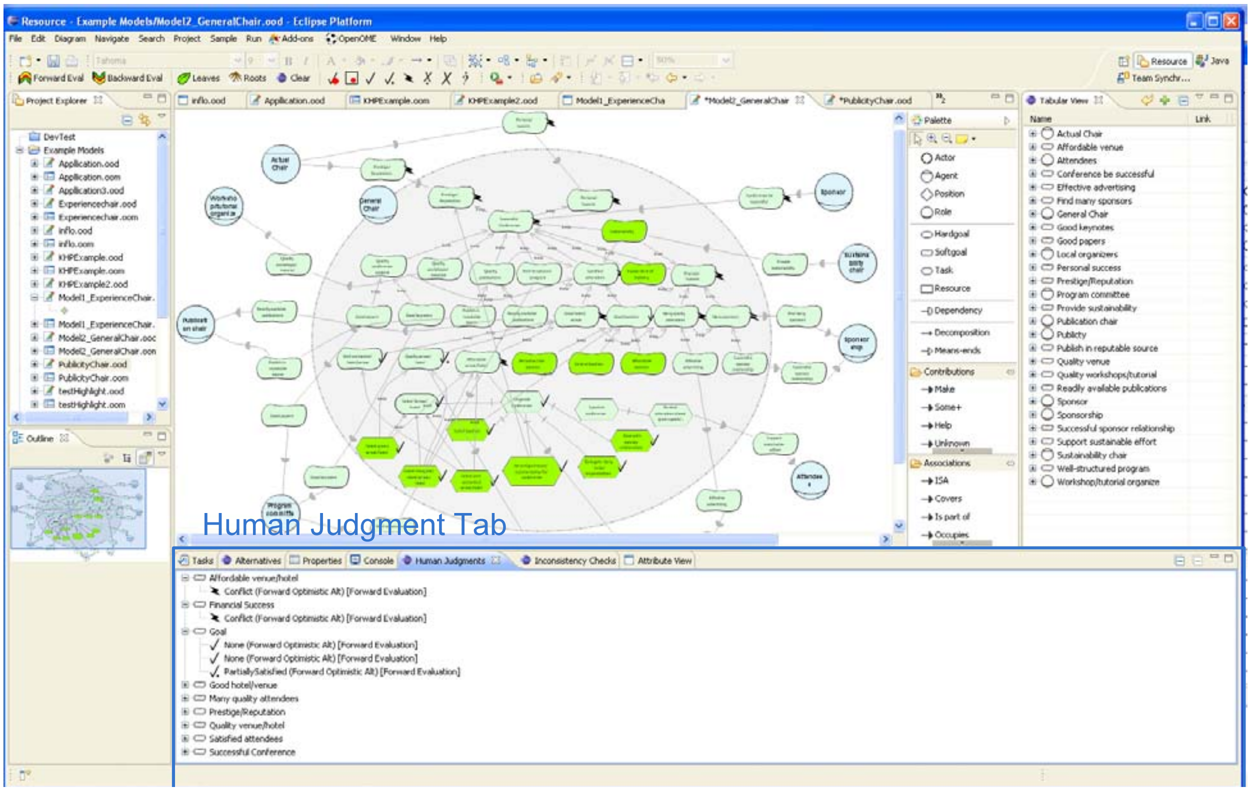


Figure 77 High-level View of Human Judgment Tab in OpenOME

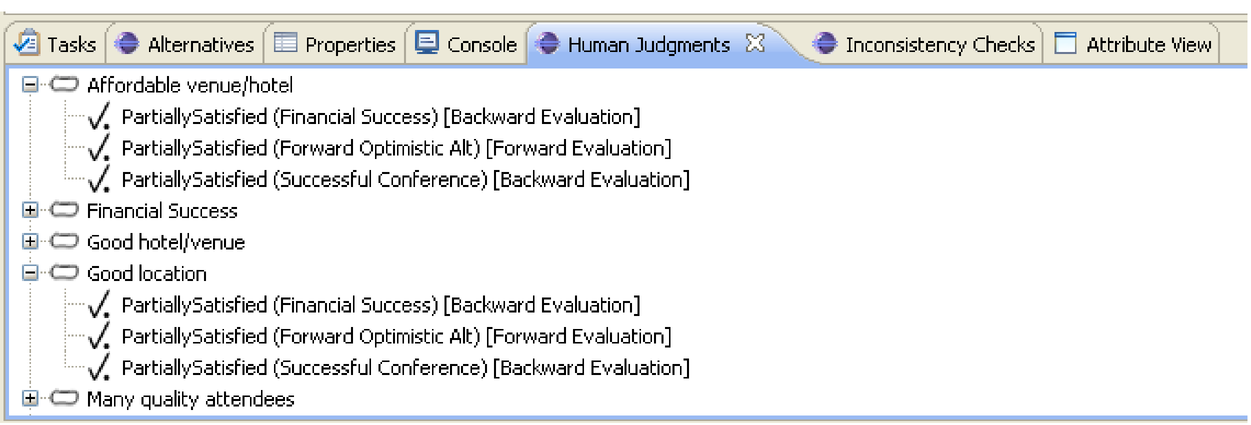


Figure 78 Detailed View of Human Judgment Tab in OpenOME

Features have been added to the Human Judgment Tab to work towards supporting model iteration. Specifically, when a judgment is changed in the Judgments Tab (via opening a judgment by double clicking on it), all analysis alternatives and respective tree slices affected by this change turn orange in the Alternatives Tab. This is intended to inform the user about the propagative effects of their judgment changes. Future work should continue this line of work to prompt users to re-evaluate affected parts of the model.

The consistency checks as described in Chapter 10 are currently being implemented as a undergraduate project (https://se.cs.toronto.edu/trac/ome/wiki/HumanJudgment). The results of these checks will be shown in the Human Judgment View, similar to results of model syntax checks.

11.2.7 Analysis Visualizations

We have described visualizations which aid analysis in Chapter 9, including highlighting leaves and roots as the starting points of analysis, highlighting intentions involved in human judgment, and highlighting intentions involved in a backward analysis conflict. Visualizations are present in the current version of OpenOME. A screenshot showing both conflict highlighting and root highlighting is shown below in Figure 79.

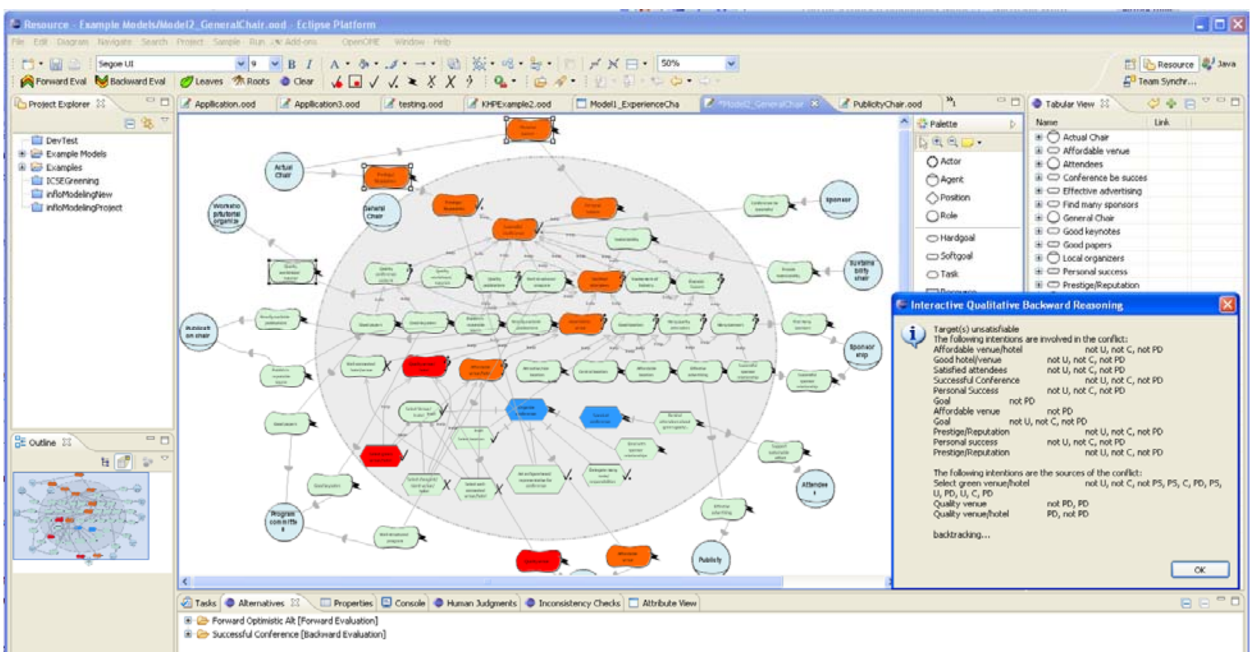


Figure 79 OpenOME Screenshot showing Conflict and Root Highlighting

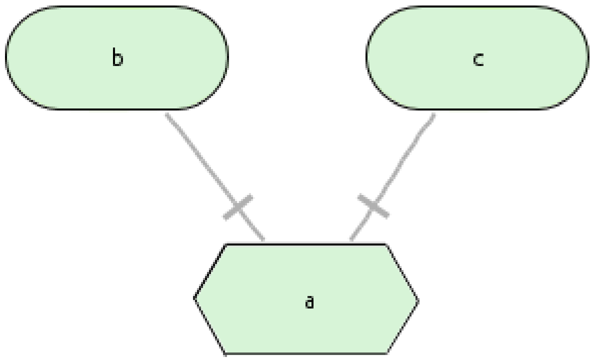
11.2.8 Syntax Checking

In Chapter 5 we have described i* syntax, including syntax checks helping to describe loose vs. strict syntax, producing user warnings or errors (Appendix A). Such syntax checks have been implemented in the current version of OpenOME. A group at University of Leipzig has created an add-on to OpenOME which exports the model into prolog and then uses SWI-Prolog to run syntax checks defined in prolog (Laue & Storch, 2011). The results of the checks are displayed in OpenOME. We have incorporated this extension into the main branch of OpenOME and added a separate view for syntax check results. Syntax checks do not yet cover all of the rules in the syntax rules list in Appendix A. Completion of syntax checking in functionality OpenOME is part of an ongoing undergraduate project. More detail on this project can be found on the Trac Wiki Project page (https://se.cs.toronto.edu/trac/ome/wiki/SyntaxChecking).

11.2.9 Importing Exporting

By default OpenOME imports and exports models in the GMF .ood and .oom format. Functionality added as part of work described in (Leite et al., 2005) has allowed for OpenOME to import models created in the textual q7 language. We have recently added functionality which allows users to export in the same language.

We are also in the progress of allowing users to import and export to and from the iStarML language (Cares & Franch, 2008; Cares, Franch, Colomer, & López, 2011). Supporting this language will allow better tool operability, enabling users to exchange models between other goal modeling tools such as jUCMNav. Figure 80 includes a simple example of an i* model and the corresponding q7 and iStarML representations. More detail on this project can be found on the Trac Wiki Project page (https://se.cs.toronto.edu/trac/ome/wiki/ImportingExporting).




```
"Do a" { &
        "b"
        "c"
}
```

```
<?xml version="1.0" encoding="UTF"-8"?>
<istarml version="1.0">
      <diagram name="My new i* diagram">
            <ielement id="AA01" name="a" type="task"/>
            <ielement name="b" type="goal">
                  <ielementLink type="decomposition" iref="AA01"/>
            </ielement>
            <ielement id="AA02" name="c" type="goal">
                  <ielementLink type="decomposition" iref="AA01"/>
            </ielement>
      </diagram>
</istarml>
```

Figure 80 OpenOME, q7 and iStarML Representations of the Same Example Model

11.2.10 Tree View

In order to alleviate scalability and comprehension issues with complex i* models, we have added a tabular tree view of the model to the OpenOME interface. The EMF/GMF implementation gives a default tree view via the .oom file type; however, this view is not customized for i* modeling (e.g., links are shown as top-level elements) and does not take into account i* circularity. A browser-based tool for creating a repository of design knowledge created a way to deal with circularity in i* models in a tree view by showing the number of instances an intention in one branch appears in other branches. Although this visualization does not relieve the complexity of circular models, it allows users to be aware of the cycles, and be aware that certain intentions appear in more than one model slice. This view is currently being implemented as part of an undergraduate project (https://se.cs.toronto.edu/trac/ome/wiki/TabularView). A detailed example view of the Tabular View for a conference planning model is shown in Figure 81.

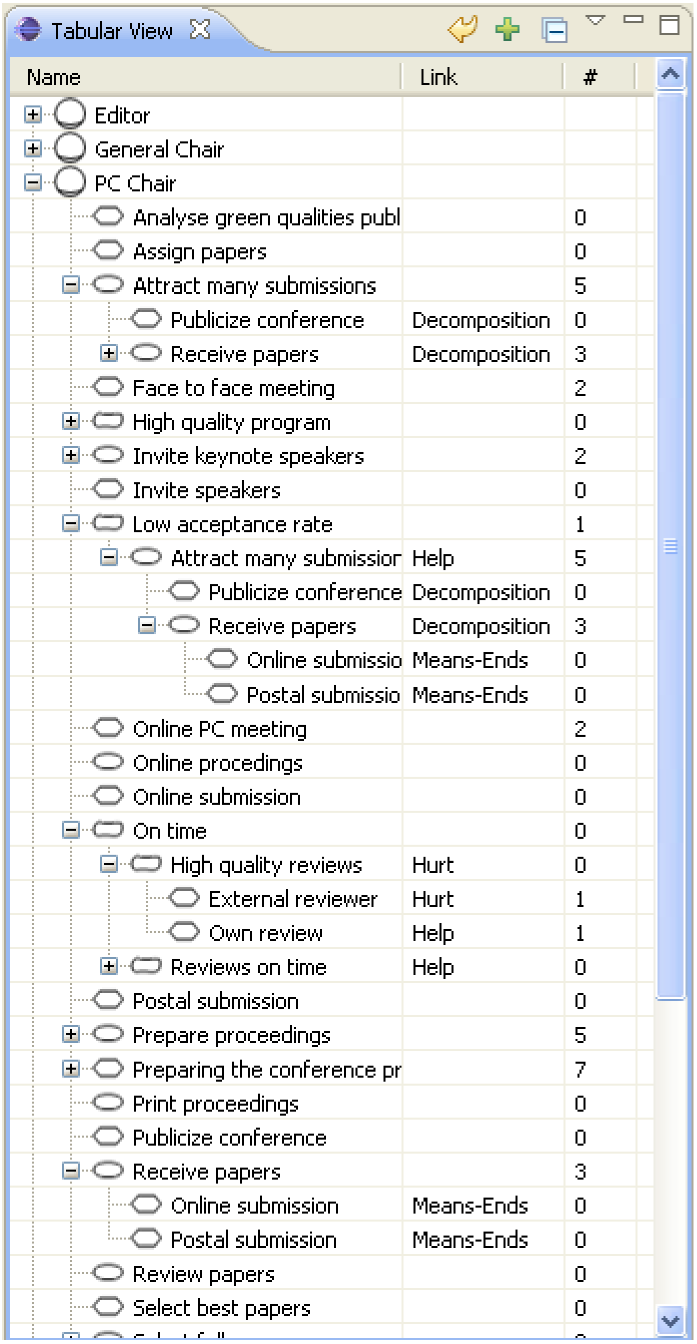


Figure 81 OpenOME Example Tabular View for Conference Planning Model

11.3 OpenOME Metamodel

The metamodel used in the OpenOME tool contains all the of the concepts and relationships needed to draw i* models as described in Chapter 5. Additional concepts are added to support interactive analysis. The parts of the metamodel used to draw i* syntax have been inherited by the initial work of Yijun Yu and Neil Ernst in building the OpenOME tool. An overview of the OpenOME metamodel is shown in Figure 82.

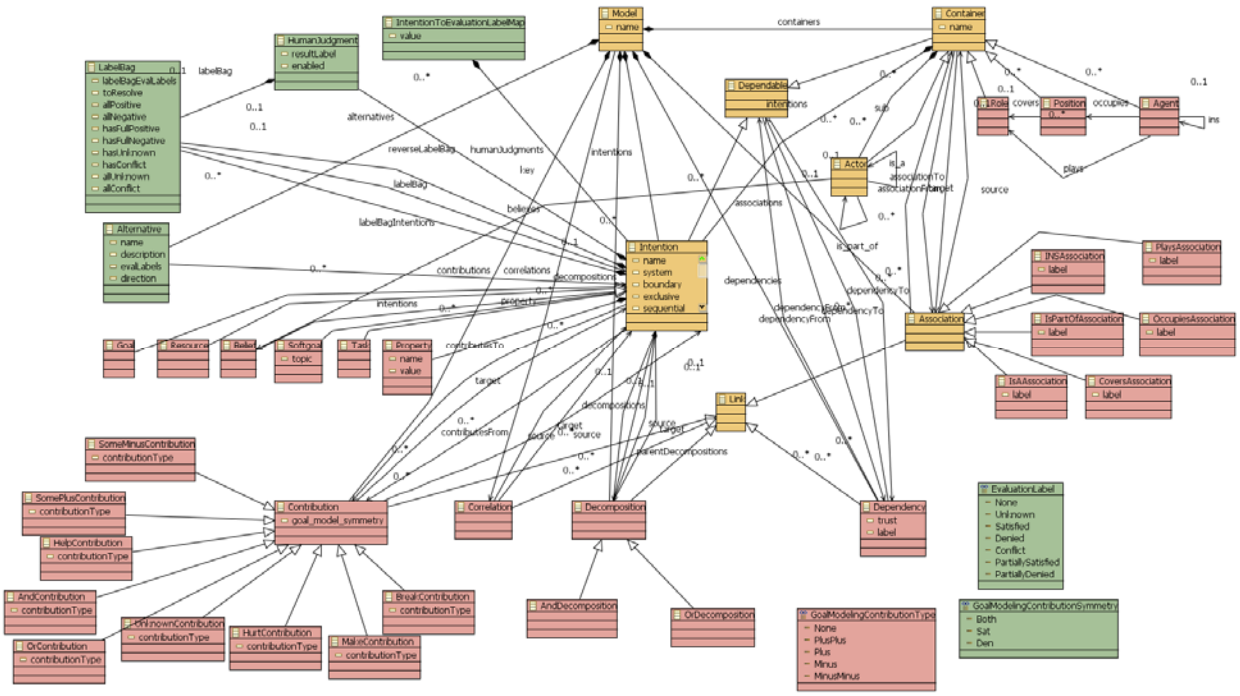


Figure 82 OpenOME Metamodel – High-level View. Color Legend: "core" i* concepts (orange), more specific i* types (red), and concepts needed for interactive analysis (green).

We can examine the concepts and relationships in the metamodel by dividing it into categories: "core" i* concepts, more specific i* types, and concepts needed for interactive analysis. In Figure 82, we distinguish between these categories using colors: orange, red, and green, respectively. Core i* types, shown more closely in Figure 83, include the Model itself, concepts which are Dependable, can be part of a dependency link, Actors which are Containers (can contain other objects), Intentions, and Associations which are Links.

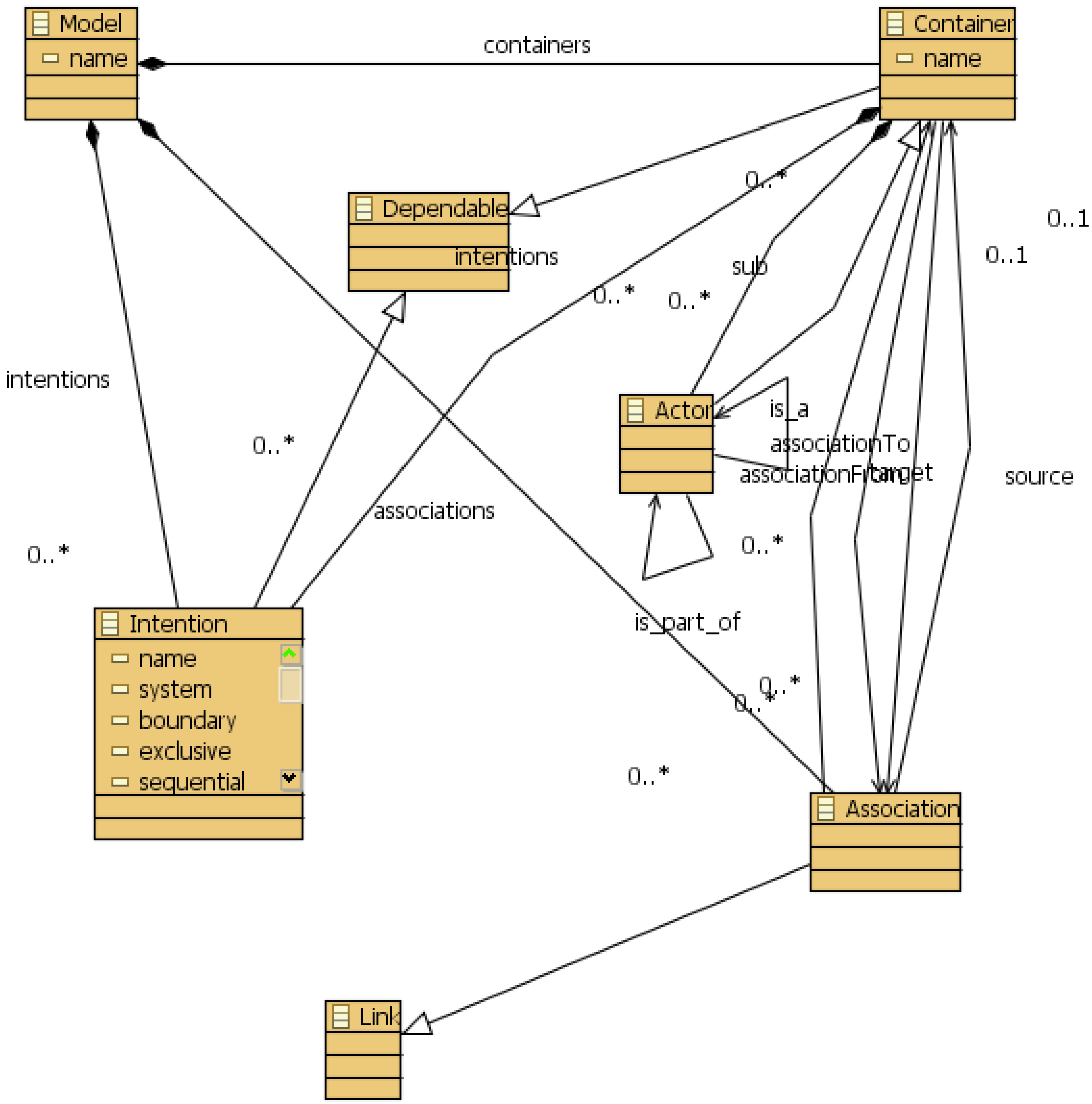


Figure 83 View of OpenOME Metamodel showing "Core" i* Concepts

Figure 84 shows an expanded view of Figure 83, dividing the core i* concepts into more specific types. Intentions are sub-classed into Goal, Resource, Belief, Softgoal, and Task. Containers can be Role, Position, or Agent. Associations are sub-classed to INS, Occupies, Plays, IsPartOf, InstanceOf, and Covers. Links are Dependency, Decomposition, Correlation, or Contribution. Contribution links are in turn broken down into their respective types.

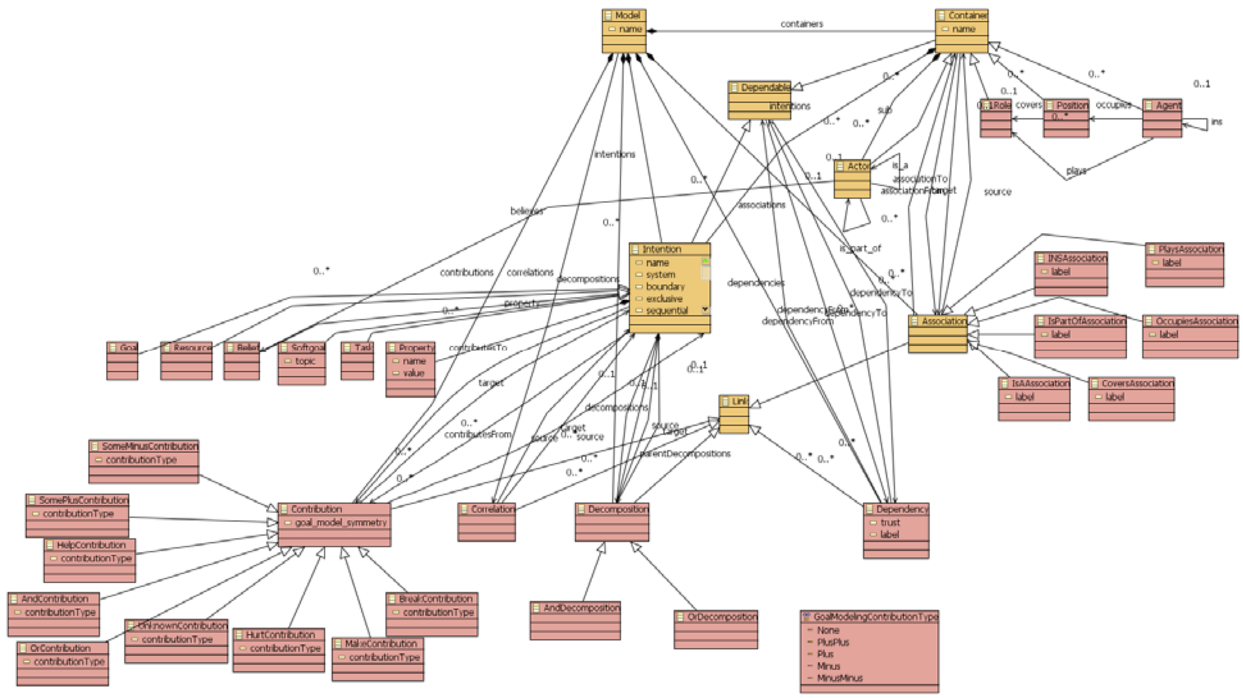


Figure 84 View of OpenOME Metamodel showing "Core" i* Concepts and more Specific Types

Figure 85 shows a closer view of the classes which implement interactive analysis. We have included Model and Intention in this model to show the relationship between analysis classes and these core concepts. Intentions have an EvaluationLabel, which is an enumerated type. A Model has 0…* Alternatives associated with it. Each Alternative associates an evaluation label with an Intention. An Intention has 0…* HumanJudgments. Each HumanJudgment has a result label and a LabelBag. Each LabelBag has a list of Intentions. Intentions also have label bags, both a LabelBag and a reverseLabelBag (for backward analysis). These bags are used to store the incoming labels before a human judgment decision is made and the LabelBags are added to a HumanJudgment object.

The metamodel includes the notion of GoalModelingContributionSymmetry, although functionality using this concept has not been added to the tool. This would allow links to propagate only negative or positive evidence, as in Tropos.

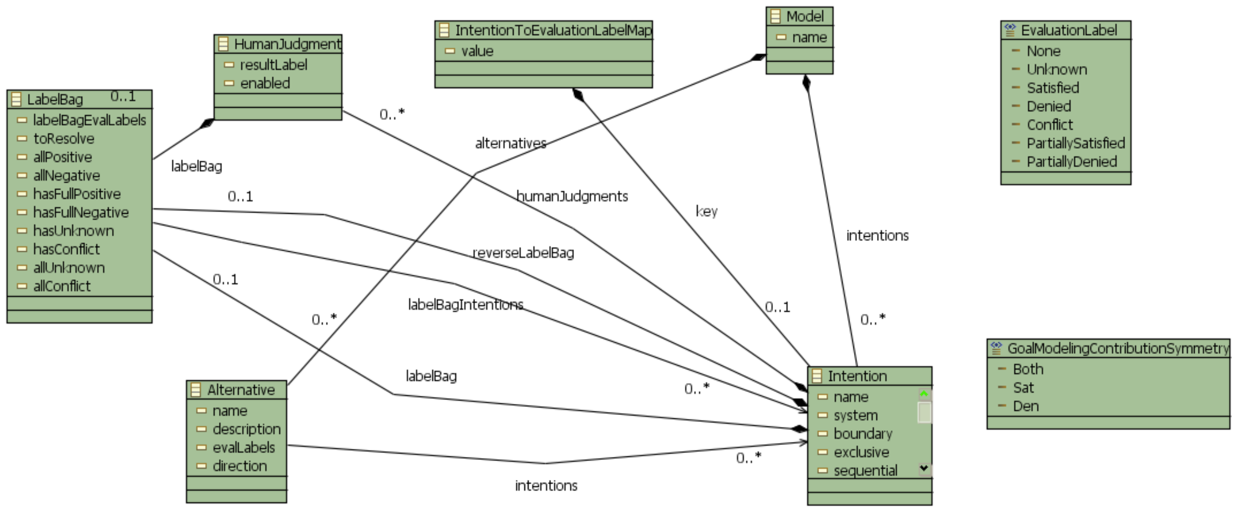


Figure 85 View of OpenOME Metamodel showing Concepts Needed for Interactive Analysis

11.4 Performance and Optimizations

In this chapter we analyze the current performance of the forward and backward algorithm implementations. We test their operation on models of a variety of sizes, and argue for a maximum practical model size for interactive early RE modeling. Current procedure optimizations are described including some optimizations over the number of human judgments. Suggestions for future optimizations are made.

11.4.1 Model Size in Practice

We have created the iterative, interactive analysis framework in this work with the early stage of the RE process in mind. Emphasis is given to maximizing the utility of stakeholder interaction to improve models and domain knowledge, given the frequent tight constraints on stakeholder time. As stated, early RE models are highly qualitative, social models which can often be difficult to formalize or quantify. As a consequence, they are difficult or impossible to generate automatically, just as a domain automatic elicitation process would not be automatable. The process of understanding and capturing domain entities and needs, to be addressed by a new system (system changes) is intrinsically human-centered. This means that early RE models must be created by hand. Even if techniques using patterns or catalogues are applied to add pre-generated parts to a model, the pattern must be manually checked for applicability and integration, a tedious manual process (e.g., Strohmaier, Horkoff, E. Yu, Aranda, & Easterbrook, 2008). Manual creation of early RE models places cognitive constraints on their size and complexity. Beyond a certain level, the models are too complicated to understand, modify or analyze effectively.

We believe that we have hit this level of complexity manually creating large i* models in our past case studies. The largest model created for the counseling service case study contained approximately 525 links and 350 elements, 230 of which represent quality criteria and system goals, the rest of which represent specific tasks in the current system (Horkoff, 2006). A high-level view of this model is shown in Figure 86. This model was created mainly by the author, and was manually analyzed, modified and interpreted by the author for the purpose of creating a requirements specification. Working with this model was cognitively difficult. It is doubtful whether anyone other than the model author would have been able to understand or modify the model without significant difficulty. Models such as this inspired our focus on scoping as part of the suggested methodology for model creation and analysis in Chapter 8.

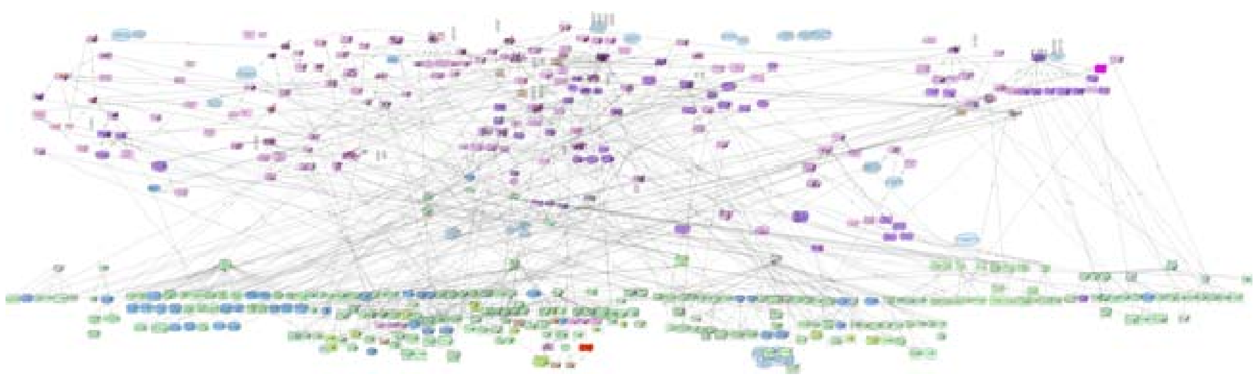


Figure 86 Very Large Counseling Service Model from (Horkoff, 2006)

Considering this model, and other similar examples, we argue that the optimal model size for domain understanding and analysis is much smaller than this size of this model. Therefore the forward and backward analysis procedures defined in this work do not need to work over very large models. The exact "optimal" size is difficult to measure precisely, and depends on many factors within the domain and the modeling experience of the modelers. In fact, the bottleneck in interactive analysis is not so much the computational complexity of the procedure, but the number of human judgments asked over a model. For large models, this number can be large, and create a very tedious process for the user. In the next section, we test the speed of the analysis implementations over several realistically-sized models created as part of actual case studies. We consider optimizations in the algorithm or human judgment collection in Section 11.4.3.

11.4.2 Scalability Tests

We test the practical run-time of our implementation over realistic examples. We choose examples of a variety of sizes and characteristics. We run two forward and two backward alternatives over each model and measure the running time of the analysis in practice. When capturing running time we differentiate between the actual computation time, and the time taken for users to read and act on various input windows, including human judgment windows and messages about conflicts in backward analysis. Tests are run on a PC with a 1.8GHz CPU and 2.5 GB of RAM (a 4-year old Dell Laptop).

We select three example models which we judge to be of small, medium, and large size, relative to our experiences in case studies and examples, described in Chapter 12. The first model is the small application model used as an example in backward analysis (Figure 34). The second model is from the conference planning example used in the exploratory experiment (more detail in Chapter 12) describing the publicity of a conference (Figure 55). The third model is the result of a group case study for the inflo modeling tool used as an example in Chapter 9 (Figure 51). Statistics concerning the three models are summarized in Table 59.

Table 59 Statistics for Scalability Test Models

Count	Model 1 Application Example	Model 2 Conference Planning	Model 3 Inflo Tool
Intentions	6	56	103
Softgoals	2	13	40
Goals		16	10
Tasks	2	27	45
Resources		0	8
Links	7	74	145
Contribution	5	35	72
Decomposition		22	15
Means-Ends	2	9	29
Dependency		8	29
Actors	1	8	12

Table 60 describes the alternatives selected in the tests for each model. We selected a mix of initial values describing sanity checks and interesting domain questions.

Table 60 Initial Analysis Labels for Scalability Tests

Model	Dir	Alt	Initial Analysis Labels
1	For	1	S(Restrict Structure of Password) and S(Ask for Secret Question)
		2	S(Restrict Structure of Password) and D(Ask for Secret Question)
	Back	1	PS(Attract Users)
		2	PS(Attract Users) and PS(Usability)
2	For	1	All leaves satisfied (S)
		2	All leaves denied (D)
	Back	1	PS(Reduce paper / glass / plastic / organic consumption)
		2	PS(High Quality Program)
3	For	1	All leaves satisfied (S)
		2	All leaves denied (D)
	Back	1	PS(Become Informed)
		2	PS(Models trustworthy) and PS(Reputation of inflo)

Table 61 and Table 62 provide the timing results from the analysis runs from the forward and backward tests, respectively. Some of the backward analysis alternatives did not produce results, either the implementation reported that there was no solution (alternative 2 in model 1), or the user gave up after several rounds of judgment (alternative 1 in model 2 and alternative 2 in model 3). In the latter cases, the implementation always reported an answer, but after several rounds of relaxing constraints for the required target, it became clear the targets could not be reasonably attained.

Table 61 Timing (Seconds) and Statistic Results for Forward Analysis Runs

Time Measures	Model 1		Model 2		Model 3	
	Alt 1	Alt 2	Alt 1	Alt 2	Alt 1	Alt 2
Num Judgments in Analysis	2	2	15	15	23	22
Num Intentions receiving Judgments	2	2	9	9	16	16
Max Judgment Time	4.109	4.875	5.813	6.390	19.734	15.078
Min Judgment Time	2.750	4.297	2.531	2.141	2.718	2.969
Average Judgment Time	3.429	4.586	4.328	3.930	8.048	6.296
Total Judgment Time	6.859	9.172	64.922	58.954	185.106	138.517
Total Computation Time	0.25	0.156	1.547	3.499	3.347	3.436
Total Analysis Time	7.109	9.328	66.469	62.453	188.453	141.953

Table 62 Timing (Seconds) and Statistic Results for Backward Analysis Runs

Time Measures	Model 1		Model 2		Model 3	
	Alt 1	Alt 2	Alt 1	Alt 2	Alt 1	Alt 2
Num Judgments in Analysis	5	3	4	2	1	5
Num Intentions receiving Judgments	2	2	1	2	1	2
Max Judgment Time	9.594	13.078	145.453	36.219	9.766	40.547
Min Judgment Time	3.047	2.062	2.032	12.813	9.766	4.438
Average Judgment Time	7.187	25.906	55.523	24.516	9.766	18.162
Total Judgment Time	35.937	8.635	222.094	49.032	9.766	90.814
Num Non-judgment Messages	2	2	4	1	1	4
Total Time for Non-judgment Messages	4.796	9.077	72.220	2.265	3.437	49.984
Total Computation Time	0.579	17.616	30.905	1.047	2.391	150.765
Total Analysis Time	41.312	35.328	325.219	52.344	15.594	291.563

Looking at the analysis results, we see that the computation time (the total time the user is waiting for an answer) for forward analysis is small (less than 4 seconds), even for larger models. As expected, the bottleneck in this form of analysis is human judgments. In the backward analysis, computational time is longer, but still manageable. Over the larger models, it can take up to 30 seconds for the tool to come up with an answer. In our experiences, this wait impedes usability to a certain degree, but the tool is still usable. In some cases, the computation time for backward analysis exceeds the judgment time, making implementation efficiency a point of interest. However, these alternatives were evaluated by the thesis author, and therefore timings for human judgments were not necessarily realistic or reflective of the interactive and collaborative aims of the procedure. We discuss potential optimizations to reduce this time in the next section

11.4.3 Potential Optimizations

The description of the forward analysis algorithm in Section 6.4.11 mentions a potential optimization when resolving a mix of hard links, could be simplified by adding attributes to intentions, or only "hard" intentions which keep track of incoming labels from a mixture of links, similar to a label bag for softgoals. However, the forward analysis procedure is not computationally complex, so such modifications would not likely make a significant difference.

As described in Section 7.5 (Further Procedure Optimizations), there are several ways to improve the efficiency of the backward algorithm. SAT solver results could be stored in a stack and used while backtracking. This may make a significant difference to the running times reported in the last section, depending on the frequency of user backtracking. Testing with other SAT solver implementation could also improve running time. Our conversion process from OpenOME models to CNF represented in Dimacs could also be optimized.

There are several ways in which the number of human judgments posed to the user could be reduced, reducing the time complexity of the procedure for the user. The most obvious option is to automatically reuse previous judgments, especially as we have already implemented the means to store and view all judgments made. However, we have avoided reusing judgments automatically across alternative analysis, as we want to allow the user to reconsider his/her judgments and optionally change his/her mind as they improve the model and learn more about the domain. The new Human Judgment view may lead us to question this implementation decision, as now users can revisit, change and delete judgments outside of an analysis run.

Further options within the Human Judgment view could provide additional alternatives which make trade-offs between efficiency and user flexibility. For example, each judgment could have the option to be automatically reused for not. In this way, users could say that they are confident in a judgment, and avoid being asked that question again. As suggested earlier in this chapter, previous judgments for an intention could be easier reused in the interface, by selecting amongst them in a list.

We have already implemented a way to derive a judgment from existing judgments, in the case where the label bag is all positive and the judgment is fully positive, the arrival of more positive evidence will not cause a new judgment to be asked (also for the converse negative case). Future work can look into ways to derive further judgments, without sacrificing user flexibility.

11.5 Conclusions

In this chapter, we have described the history, features, and metamodel of the OpenOME tool supporting i* modeling and interactive i* analysis. The tool hides much of the complexity (R16) of interactive analysis, providing partial automation (R3). We have performed tests to ensure the scalability of the analysis implementation over reasonably-sized models (R1). The contributions of this chapter are summarized in Figure 87.

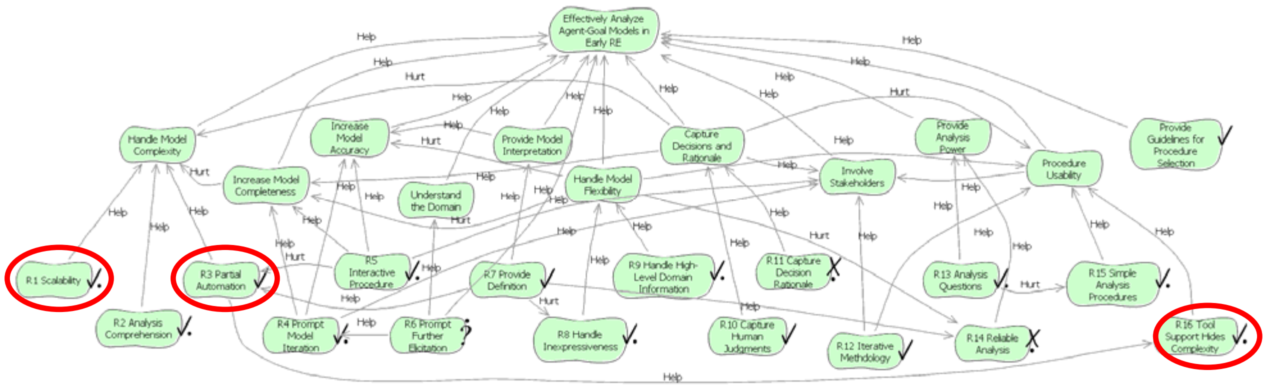


Figure 87 Focus of the Current Chapter in the Requirements for analysis of Agent-Goal Models in Early RE

Development on general tool functionality and bug fixes for the OpenOME tool is ongoing. Use of OpenOME, including the forward and backward evaluation procedures and current visualizations, has been tested in a series of user studies, described as part of the next chapter.

Chapter 12 Framework Validation

Most of the research on analysis procedures focuses on the analytical power and mechanisms of the various analysis procedures, typically demonstrating utility by providing a single example application, often in the context of an industrial project. To our knowledge, little work has been done to study how modelers analyze goal models – to compare ad hoc analysis (without a systematic procedure) with the application of proposed procedures. Without a systematic analysis procedure, the modeler/analyst may be examining the model in an ad hoc manner, possibly mixing forward and backward propagation of values, or assigning values to model intentions without following a predetermined systematic process.

In this work, we have provided goal model analysis procedures specifically suited to early stages of RE system analysis, supporting qualitative analysis over imprecise concepts, and encouraging iteration over models and elicitation over the domain. The framework described in this thesis has been validated in several different studies. We will review studies which have applied an initial version of the forward analysis described in this thesis described in (Horkoff 2006; Horkoff et al. 2006; Easterbrook et al. 2005). These studies include the early stages of the counseling service example used in the introduction, and a study of Trusted Computing Technology. As part of this work, the forward analysis procedure has been applied to further stages of the counseling service study, including the analysis of patterns and knowledge transfer agents. Results from these studies have indicated that such analysis increases model iteration, prompts further elicitation, and improves domain knowledge.

A small exploratory experiment was conducted to test the perceived benefits of the forward procedure, namely:

Analysis: aids in finding non-obvious answers to domain analysis questions (R13),

Model Iteration: prompts improvements in the model (R15),

Elicitation: leads to further elicitation of information in the domain (R6), and

Domain Knowledge: leads to a better understanding of the domain (R6).

This chapter is an expansion of the following papers/reports:

Horkoff, J., Yu, E., & Liu, L. (2006). Analyzing Trust in Technology Strategies. In International Conference on Privacy, Security and Trust (PST 2006) (pp. 21-32).

Strohmaier, M., Yu, E., Horkoff, J., Aranda, J., & Easterbrook, S. (2007). Analyzing Knowledge Transfer Effectiveness – An Agent-Oriented Modeling Approach. Proceedings of the 40th Annual Hawaii International Conference on System Sciences (p. 188b). IEEE Computer Society.

Strohmaier, M., Horkoff, J., Yu, E., Aranda, J., & Easterbrook, S. (2008). Can Patterns Improve i Modeling? Two Exploratory Studies. Proceedings of the 14th International Working Conference on Requirements Engineering Foundation for Software Quality (pp. 153-167). Springer.*

Horkoff, J., & Yu, E. (2009a). Evaluating Goal Achievement in Enterprise Modeling – An Interactive Procedure and Experiences. The Practice of Enterprise Modeling (pp. 145-160). Springer.

Horkoff, J., & Yu, E. (2010b). Interactive Analysis of Agent-Goal Models in Enterprise Modeling. International Journal of Information System Modeling and Design (IJISMD).

Horkoff, J., & Yu, E. (2010d). Interactive Goal Model Analysis Applied - Systematic Procedures versus Ad hoc Analysis. The Practice of Enterprise Modeling, 3rd IFIP WG8.1 (PoEM'10).

Easterbrook, S., Yu, E., Aranda, J., Horkoff, J., Strohmaier, M., Fan, Y., Leica, M., et al. (2011). Analyzing Requirements for Online Presence Kids Help Phone. iStar Showcase.

Horkoff, J. (2011a). Using i Modeling for the Analysis of Strategy Documents: Course Project for CS2507 Conceptual Modeling, Spring 2007. DCS Technical Report, CSRG-613.* ftp://ftp.cs.toronto.edu/csrg-technical-reports/613/

Horkoff, J. (2011b). Applying the i Framework, Scenarios, and Prioritization Methods to a Requirements Analysis for Kids Help Phone: Course Project for CS2106 Requirements Engineering, Fall 2005. DCS Technical Report, CSRG-612.* ftp://ftp.cs.toronto.edu/csrg-technical-reports/612/

Horkoff, J., Yu, E. (2011d). Modeling and Analyzing Technology Strategies. DCS Techncial Report, CSRG-614. ftp://ftp.cs.toronto.edu/csrg-technical-reports/614/

Results of this study did not provide strong evidence to support previous claims, showing that benefits, when they occur, can occur both with systematic and ad-hoc model analysis. The last two hypothesis, concerning elicitation and domain knowledge proved to be difficult to test empirically. Although we believe that the interactive, iterative procedures designed in this work will have a positive effect on prompting elicitation and increasing domain knowledge, in our requirements for early RE analysis we focus on more measurable effects, such as increasing model completeness and accuracy. Based on the results of the initial exploratory studies, further exploratory and confirmatory studies were designed.

Two types of case studies were designed and administered to further test the hypothesized benefits of interactive analysis. Due to the great number of confounding variables, we chose to use case studies as the research method, rather than experiments producing statistically significant data. Specifically, we conducted ten case studies using subjects with some experience in i* modeling. Half of the participants analyzed models using no explicit procedure (ad-hoc analysis) while the other half used implementations of the forward and backward interactive analysis procedures.

Previous work hypothesized that interactive analysis provokes useful group discussions. In order to gain some insight into analysis by individuals versus analysis in a group, we administered a separate multi-session case study involving a project team designing tool support for modeling "back of the envelope" calculations.

Five follow-up studies were conducted using the individual participants in order to test the usability and effectiveness of the visualizations described in Chapter 9.

Qualitative and quantitative analysis of results were used to compare treatments in both studies, to gather evidence to support or deny the hypotheses, and to gain an understanding of the benefits of and barriers to systematic goal model analysis, helping to guide the application of goal analysis for systems within an organization.

Results show that both ad-hoc and systematic analysis can provoke iteration, elicitation, and improve domain knowledge. However, the results reveal other benefits of systematic analysis, such as a more consistent interpretation of the model, more complete analysis, and the importance of training.

Our empirical studies aim to test how the forward and backward procedures satisfy several of the requirements for early RE agent-goal model analysis. We have already explored the computational scalability of the procedure, but by performing several studies which test the procedures in practice, we test the practical scalability of the procedure (R1), including users' ability to comprehend analysis (R2). These studies also test the practical simplicity of the analysis procedures (R15), including the ability of the tool support to hide complexity (R16). By using the procedures in practice, we test their ability to answer analysis questions in the domain (R13). We also test the ability of the procedure to prompt model iteration as compared to ad-hoc analysis (R4), and to prompt further elicitation (R6), improving domain understanding. The last rounds of studies tested the usability of the iterative methodology described in Chapter 8 (R12). Goals addressed by this chapter are highlighted in Figure 88.

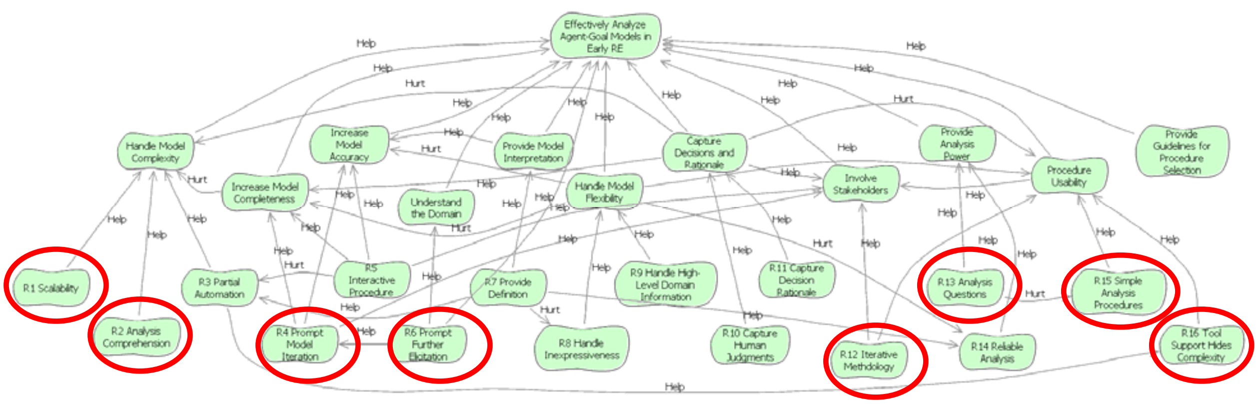


Figure 88 Focus of the Current Chapter in the Requirements for analysis of Agent-Goal Models in Early RE

12.1 Manual Application of Forward, Interactive Analysis

The concepts and algorithm associated with the forward analysis procedure were created and applied before the procedure was implemented in OpenOME. Some parts of these studies have been described as part of (Horkoff, 2006). Others were conducted after this work was complete. We describe the use of analysis in several such studies, including initial exploratory studies, the Trusted Computing, the Strategy Document, and the counseling service case study. Results lead to the design and execution of an exploratory experiment testing the perceived benefits of forward analysis.

12.1.1 Initial Exploratory Examples

Initially i* modeling was used to model and analyze domains with textual documents as sources. Models were created to analyze the Montreux Jazz Festival, based on business models created by (Osterwalder, 2004). Models were created to analyze E-Commerce data exchanges, as described by (Spiekermann, Dickinson, Günther, & Reynolds, 2003), and Economic Information Security as described by (Anderson, 2001). More detail concerning these studies can be found in (Horkoff, 2006). Initially, these studies applied the qualitative, interactive evaluation procedure described as part of the NFR procedure, with adjustments and additions made for the agent-oriented concepts in i*. Experiences showed that this procedure did not provide enough flexibility due to too much automation. For example, all softgoals receiving conflicting evidence were automatically assigned a conflict label. These issues have been explored in Section 2.3.1.2 and Chapter 4. The high-level and informal nature of the analysis domains called for a procedure with more user control. Adjustments to the role of human judgment were made, and a more detailed description of the forward procedure was produced, as described by (Horkoff, 2006) and re-described formally with minor modifications in Chapter 6. The next steps involved testing the procedure on larger, more complex, case studies.

12.1.1.1 Study Contributions

Qualitative, interactive evaluation was helpful in getting more value out of agent-goal models, especially concerning domain understanding and model improvement, but flexibility in combining conflicting evidence and a clearer procedure description was needed.

12.1.2 Trusted Computing

The forward analysis procedure described in this work was applied to a case study analyzing the effects of Trusted Computing technology. This work was performed from 2005 to 2007. Earlier versions of the study were reported in (Horkoff, 2006), with later descriptions appearing in (Horkoff, E. Yu, & Liu, 2006) and (Horkoff & E. Yu, 2011d). The aim of this study was to use i* modeling and analysis to understand the claims behind Trusted Computing (TC) Technology, given that it was a controversial technology about which a variety of seemingly contradicting claims were made. The approach used the TC example as a means to analyze technology design in relation to business strategies. The study used modeling and forward analysis to describe the differences between viewpoints concerning the effects of TC technology in relation to organizational strategy. Sources for the models were vendor documents, blogs and news articles relating to the new technology.

12.1.2.1 Background: TC

Trusted Computing (TC) refers to technology, applicable to personal computers and other personal electronic devices, which has been proposed by a set of technology vendors, now represented by the Trusted Computing Group (TCG). The proponents of this technology have claimed that it will promote security for the average user while not preventing the use of pirated content. However, the parties who are opposed to the technology claim that it will in fact give control of technology to technology vendors, effectively threatening security. Trusted Computing opponents claim that the primary motivation for the technology is to combat software piracy and further implement digital rights management (DRM).

12.1.2.2 Method

The modeling and analysis were driven by a series of domain questions, such as: What does the Technology User want? What makes technology trustworthy? What does the Technology Provider do? How does this relate to Licensed/Copyrighted Content? How is Licensed/Copyrighted Content provided? Modeling started by capturing the business of content and technology, the context of the model which did not appear to differ between viewpoints. Forward analysis over this model showed a tension between consumer's desire for affordable products and the technology provider's desire for higher profits. The next stage focused in the introduction of malicious parties (What does the Data Pirate have to offer?, What is the effect of Hacker/Malicious Users?) using analysis to show how these players facilitated goals of the Technology User while threatening goals of the Technology Providers and Licence/Copyright Owner. The next steps involved depicting the effects of Trusted Computing from both the proponent and opponent perspectives, comparing the model and analysis results. From the proponent viewpoint, we asked: How does Trusted Computing help? Will TC Work? Forward analysis was used to show that trusted computing would satisfy the goals of the Technology Providers which make products desirable, (products provide security, privacy, compatibility, etc.), making it a viable technology (Figure 89). In the opponent viewpoint we ask: What do opponents say about Trusted Computing? and What are the overall effects of Trusted Computing? From this viewpoint, TC locks customers in, thwarting data piracy, while poorly protecting against hackers/malicious users (Figure 90).

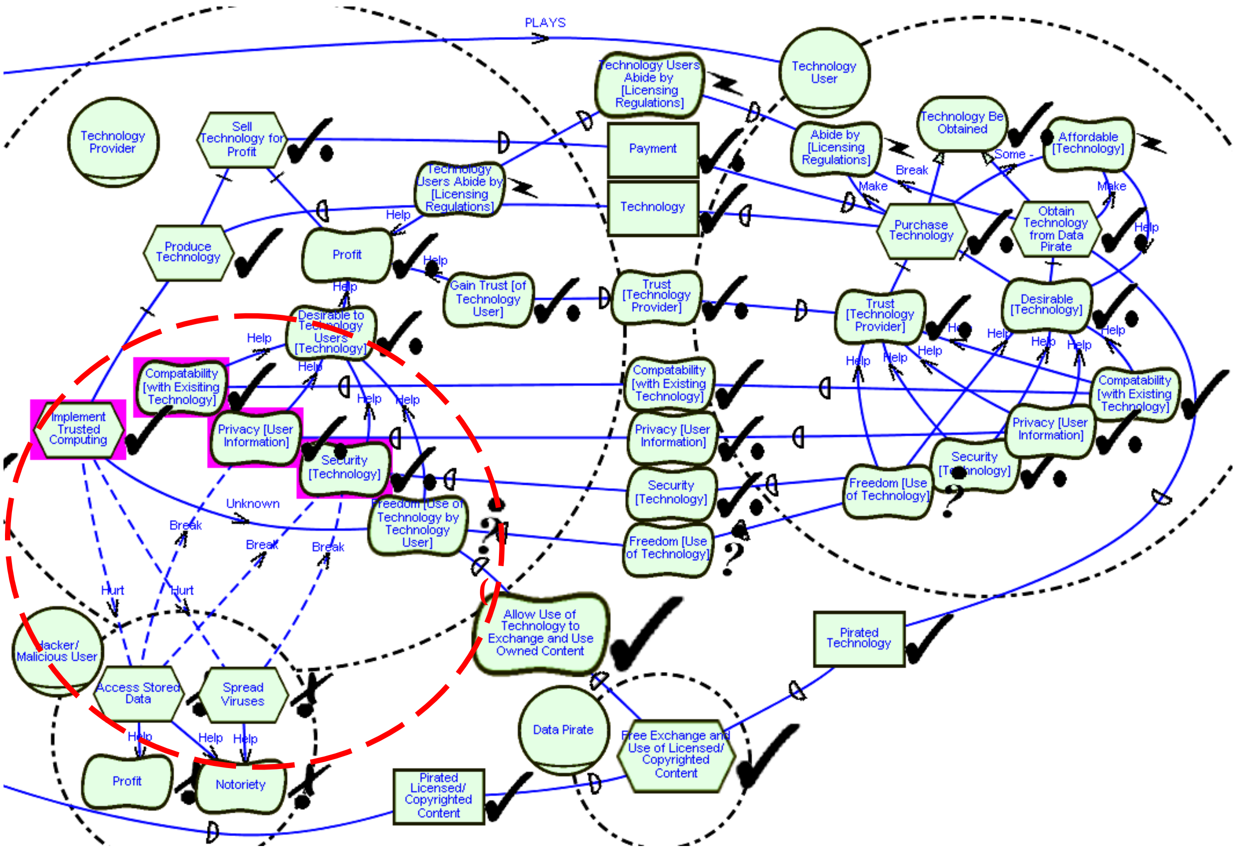


Figure 89 The Effects of Trusted Computing According to Proponents

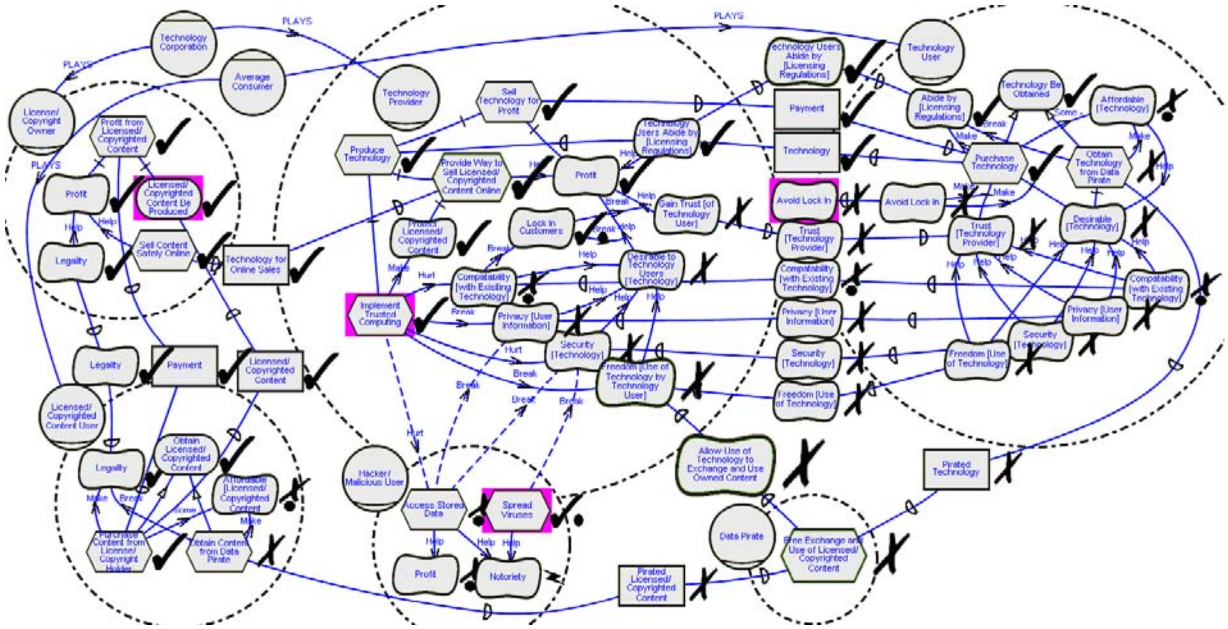


Figure 90 The Effects of Trusted Computing According to Opponents

12.1.2.3 Study Contributions

The Trusted Computing study applied forward analysis manually to medium to relatively large (see Figure 90) i* models. This helped to show the cognitive scalability (R1) and analysis comprehension of the procedure (R2), although the analysis was only performed by one person, the thesis author. The TC study demonstrated the ability of the procedure to provoke further elicitation and subsequent model iteration (R4). For example, although the model appeared to be sufficiently complete, one of the first rounds of analysis of the TC Opponent point of view revealed that Technology Users would not buy TC Technology. Although this may be the case for some users, obviously the makers of TC Technology envisioned some way in which users would accept their product. These results led the modeller to further investigate the sources, including factors such as product lock-in, more accurately reflecting the domain.

The TC example attests to the simplicity of forward analysis (R15), although one of the conclusions of the study was that tool support providing some level of automation was needed. The study analysis demonstrated the procedure's ability to accommodate high-level domain information such as security and privacy, drawing conclusions over these types of concepts (R9). Example analyses over models demonstrated that the procedure can be used to answer useful analysis questions over the domain (R13), such as How do malicious parties affect the equilibrium between the TC Producer and User? and Will TC Work (proponents)? Of course such questions could be asked and answered without a model or analysis, but creating a model provides a visual summary over which to share perceptions and agree on the scope of the domain. As described in Section 6.1, systematic analysis over such models becomes necessary when models become complex, allowing evaluators to make their arguments concrete using sharable visual artefacts.

12.1.3 Strategy Document

As part of a course project, the author conducted a case study examining the ability of i* modeling and forward analysis to represent and evaluate strategy documents. Although manual forward analysis was applied to some parts of the examples, the focus was on the benefits of i* modeling for strategy document analysis, and not specifically on the application of interactive analysis. We summarize the case study findings below.

12.1.3.1 Motivation

Often, for various reasons, it becomes necessary for an organization to produce a document outlining strategies and plans which direct the future of the organization. Such documents usually contain a description of the objectives of the organization; including the actions which the organization plans take in order to meet its objectives.

Due to the presence of multiple, sometimes conflicting motivations for the creation of strategy documents, such documents can be difficult both to interpret and to create. From the point of view of the reader, such documents can suffer from several problems including a general difficulty in understanding the content, often due to the complex and extravagant language used; confusion as to how exactly each objective will be accomplished, or to what objective each part of the plan aims to address; and an uncertainty as to how to assess the progress an organization may have made in accomplishing the plan. Furthermore, although some of the intrinsic problems in such documents may be apparent to document readers, others may go unnoticed without a more in-depth evaluation. Strategy document readers may be too caught up in understanding the details of the document to ask potentially important “how?” or “why?” questions, or to notice potential conflicting objectives.

For an example which potentially highlights some of these problems, we can examine several related excerpts from an Academic Strategy document for the Faculty of Information Studies (FIS).

> "β. At the level of **information practice**: take a leadership role in establishing a wide range of strategic partnerships…"
>
> <Two more paragraphs here>
>
> "Understanding how the β-part fits into the mission requires understanding a university's overall information strategy in terms of 3 interrelated levels:"
>
> "L3. …"
>
> "L2. *Information* **practices**: An intermediate level of socio-technical information practices, including publication, peer review, libraries, student work, financial and administrative services, etc."
>
> "L1. …"

What conceptual map would readers develop of the strategic plan after reading these sections of the document? What questions would they have? Perhaps they may ask "Partnerships with whom?" or "What is an information strategy?" or "Are the two sections concerning information practices consistent with each other?". Do these questions get to the root of potential confusion, or are important questions being missed? Is there a way to aid the user in discovering and effectively expressing useful questions concerning these and other document excerpts?

12.1.3.2 Hypotheses

This study used i* modeling and manual forward qualitative analysis to capture and understand strategy documents. The objectives of the study were:

- To formulate the apparent synergistic benefits of i* modeling with strategy document analysis and creation in a series of exploratory hypothesis. Specifically:
    - **Document Comprehension.** Creating i* models describing strategy documents can help to facilitate understanding of the strategy, including clarifying goal relationships.
    - **Strategy Analysis.** Modeling and analyzing strategy documents can help to evaluate strategy achievability, find hidden contradictions, reveal vulnerabilities, and assess plan progress.
- To determine the importance of tool support for this type of analysis.

- To create a serious of guidelines and recommendations for future application of i* modeling to strategy documents.

12.1.3.3 Method

The viability of the hypotheses was tested using two exploratory case studies, each focusing on creating models reflecting a different strategy document. An attempt was been made to select subject documents which differ in style and content. The focus of the first case study was on an academic plan for FIS (Faculty of Information Studies) at the University of Toronto, written in 2004 (Smith, 2004). The second case study looks at the National Security Strategy (NSS) for the United States of America, produced in 2006 (*The National Security Strategy of the United States of America*, 2006). For both documents, i* models were created to summarize the strategies within. Scalability issue in both cases were prominent. The models became large and unwieldy after only a small portion of the documents were modeled. Strategies to deal with scalability by assigning model sections to sections of the document, or to different categories were devised. Models were created in multiple files and merged together. A high-level view of the merged model created for the FIS Strategy document is shown in Figure 91. Purple boxes represent semantic groupings. More details, included the created methodology for modeling strategy documents, can be found in (Horkoff, 2011a).

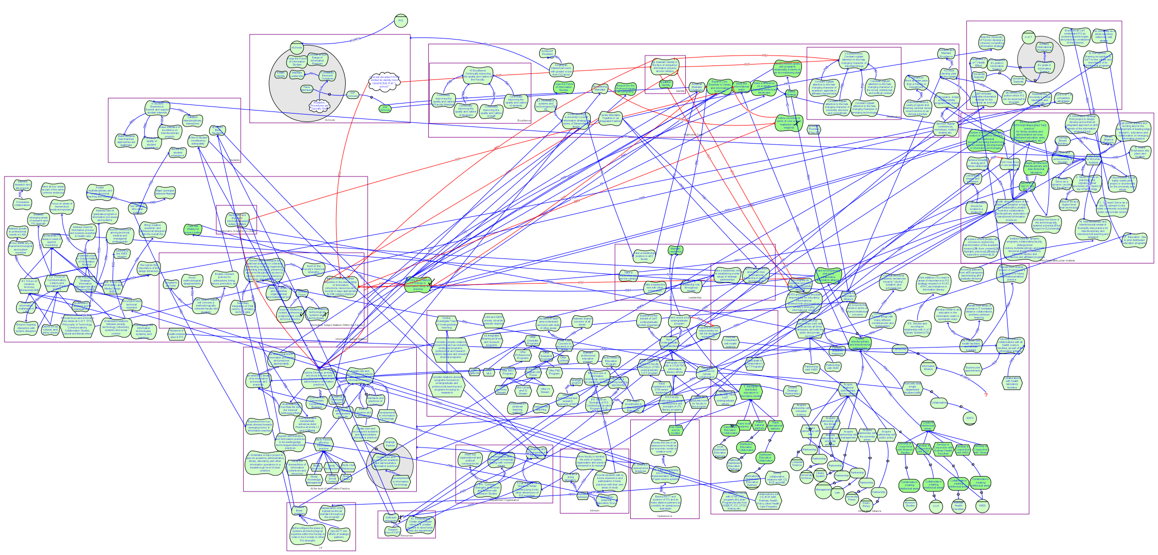


Figure 91 High-Level View of All-Encompassing Model for the First 15 Pages of the FIS Strategy Document

12.1.3.4 Results

Overall, analysis of the case study results found that i* modeling and analysis was helpful in facilitating analysis and clarifying goal relationships. Several clarification questions were derived from the modeling exercise. Because the models representing the documents were not complete, and were complex, we were reluctant to apply qualitative analysis to the models. Very complex models such as the one in Figure 91 were not analyzed directly. However, forward analysis was applied to parts of this model, before merging (Figure 92), and to parts of the NSS document (Figure 93). Models labels were applied manually using a template in Microsoft Visio.

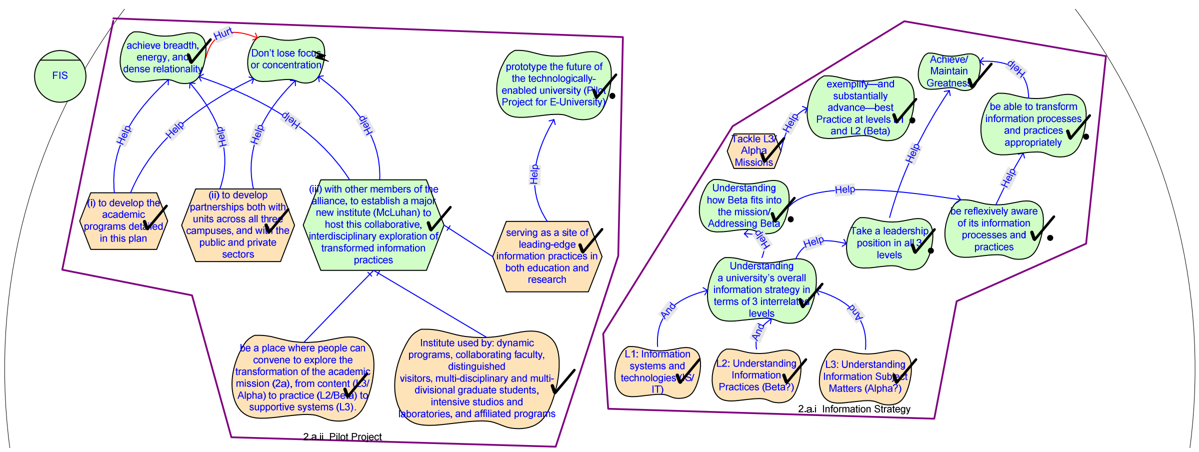


Figure 92 Evaluation of the Model Representing Sections 2ai and ii of the FIS Document

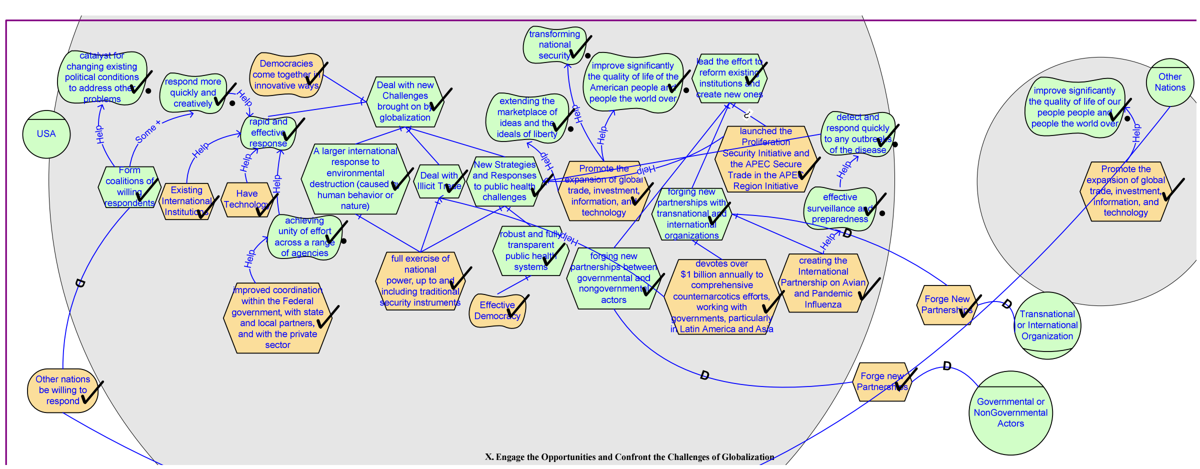


Figure 93 Evaluation of the Model Representing Section 10 of the NSSUSA Document

Qualitative analysis of the results using a collection of examples showed that forward analysis was useful in determining whether strategies were achievable, even though each individual model was not complete as per the document. Generally, on a local level, most elicited goals were at least partially achievable given the tasks described in the strategy, as can be seen in Figure 92 and Figure 93. The modeling process was also seen to be useful in finding hidden contradictions and revealing vulnerabilities. It was found to be too difficult to assess progress without the availability of some sort of progress report. However, forward analysis was not used extensively in the testing of these hypotheses. In addition to the initial hypothesize benefits, case study experience lead to several ideas concerning the use of i* models to reflect document structure and to aid in an assessment of document organization.

12.1.3.5 Study Contributions

As this work was undertaken in an early stage of the development of the framework described in this thesis, hypotheses such as the ability of the analysis to prompt model changes were not explicitly tested. However, application of forward analysis revealed that i* models created from documents were not well connected. In other words, there were many isolated clusters of elements which could not be related through evaluation to other clusters. In the strategy documents, such discoveries would lead the modeller to further extrapolate relationships which are not present in the text to make the model more complete. In general, if the sources of the models, such as stakeholders in an early RE process, are available, such discoveries may lead to model improvement (R4) and further elicitation. The study also demonstrates the procedures ability to answer analysis questions (R13), in this case, is the strategy as described locally achievable?

12.1.4 Counseling Service

The counseling service study applied agent goal modeling and analysis to a domain involving a real enterprise, including access to a variety of stakeholders. This case study, used in several of our examples, applied the forward evaluation procedure in several stages of the project. The first two stages of the project were described and used as an illustrative case study in (Horkoff, 2006). Additional details can be found in Horkoff (2011b). The third stage of the case study was conducted after, used to further test the qualitative forward procedure.

12.1.4.1 Background

Kids Help Phone is a not-for-profit organization that has provided phone counseling for Canadian youth since 1989. After gaining nation-wide recognition as for its services, Kids Help Phone made the transition to online counseling, starting in 2002. This transition brought with it critical considerations for the organization. Online counseling could be viewed by multiple individuals, and may provide a comforting distance which would encourage youth to ask for help. However, in providing counseling online, counselors lose cues involved in personal contact, such as body language or tone. Furthermore, there were concerns with confidentiality, protection from predators, public scrutiny over advice, and liability over misinterpreted guidance. How could such an organization explore and evaluate options for online counseling, balancing the needs of multiple parties?

The collaborative research project between Kids Help Phone and investigators within the Bell University Labs at the University of Toronto was launched in 2004 to perform a strategic analysis of the information needs of Kids Help Phone, in light of their increased use of and dependence on technology, to facilitate and support their counseling process. The research goals of this project evolved throughout its lifetime to fit the needs of Kids Help Phone, by addressing those issues that were deemed prominent in interviews and meetings with Kids Help Phone staff. The strategic requirements management project ended in 2008 after several stages.

12.1.4.2 Project Stages and Lessons Learned

Stage 1. The first stage of the project tested the application of i* modeling to a large organization. Manual i* models were created to describe various aspects of the organization (the largest had 353 elements). These models were used along with the forward qualitative i* evaluation procedure (Horkoff, 2006) in order to analyze and compare the effectiveness of technology options for providing counseling over the internet. Figure 94 and Figure 95 show a zoomed in and out view, respectively, of an example analysis alternative over a model representing counseling services. The models were created based on transcripts of interviews with several roles in the organization. The results were presented to the organization using reports and presentation slides containing small excerpts of the model. The analysis was well-received by the organization, bringing to light several issues and provoking interesting discussion. The organization opted to continue to use a modified version of the moderated bulletin board option already in place, due partially to a lack of resources available to handle online counseling traffic. Results of this study were used to study the use of viewpoints in conceptual modeling (Easterbrook et al., 2005).

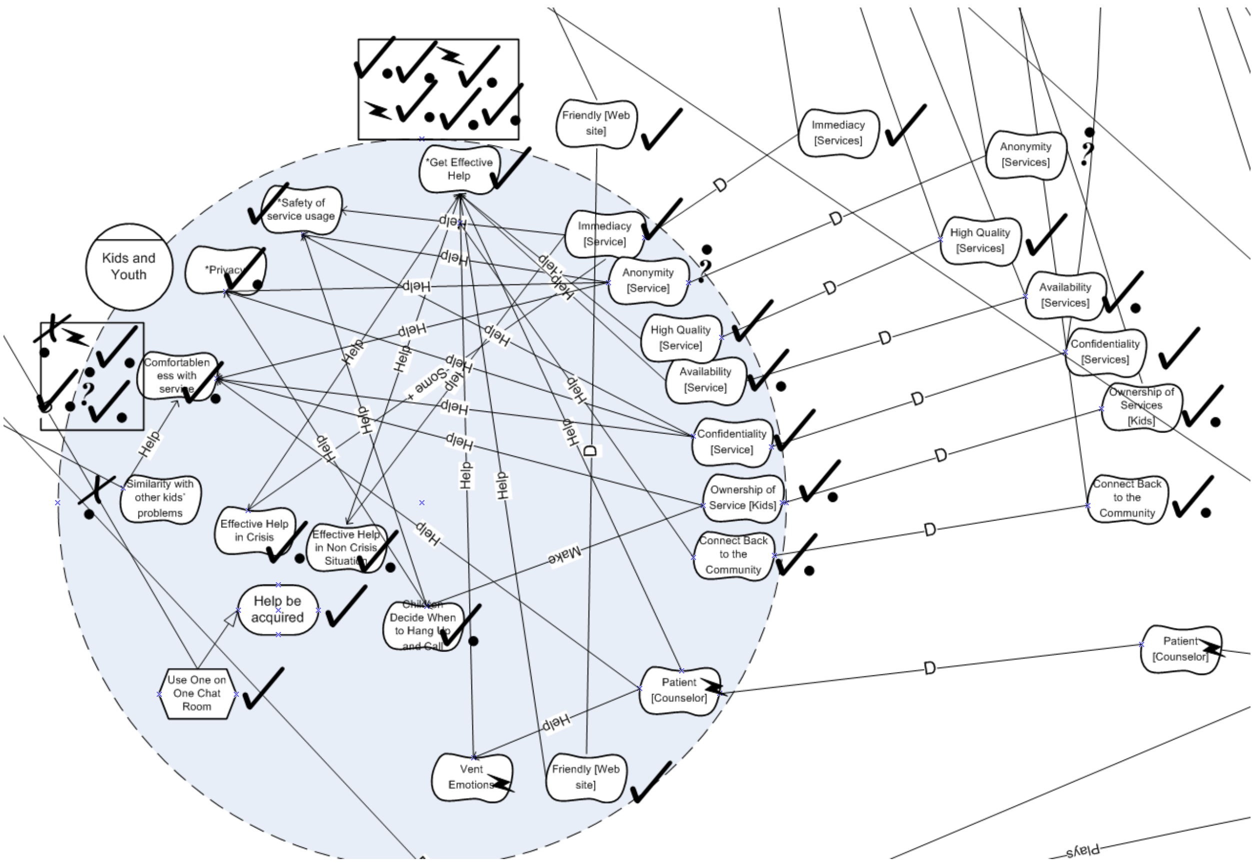


Figure 94 An Excerpt from a Counseling Service Model Analysis Alternative Evaluating One-on-one Chat Rooms

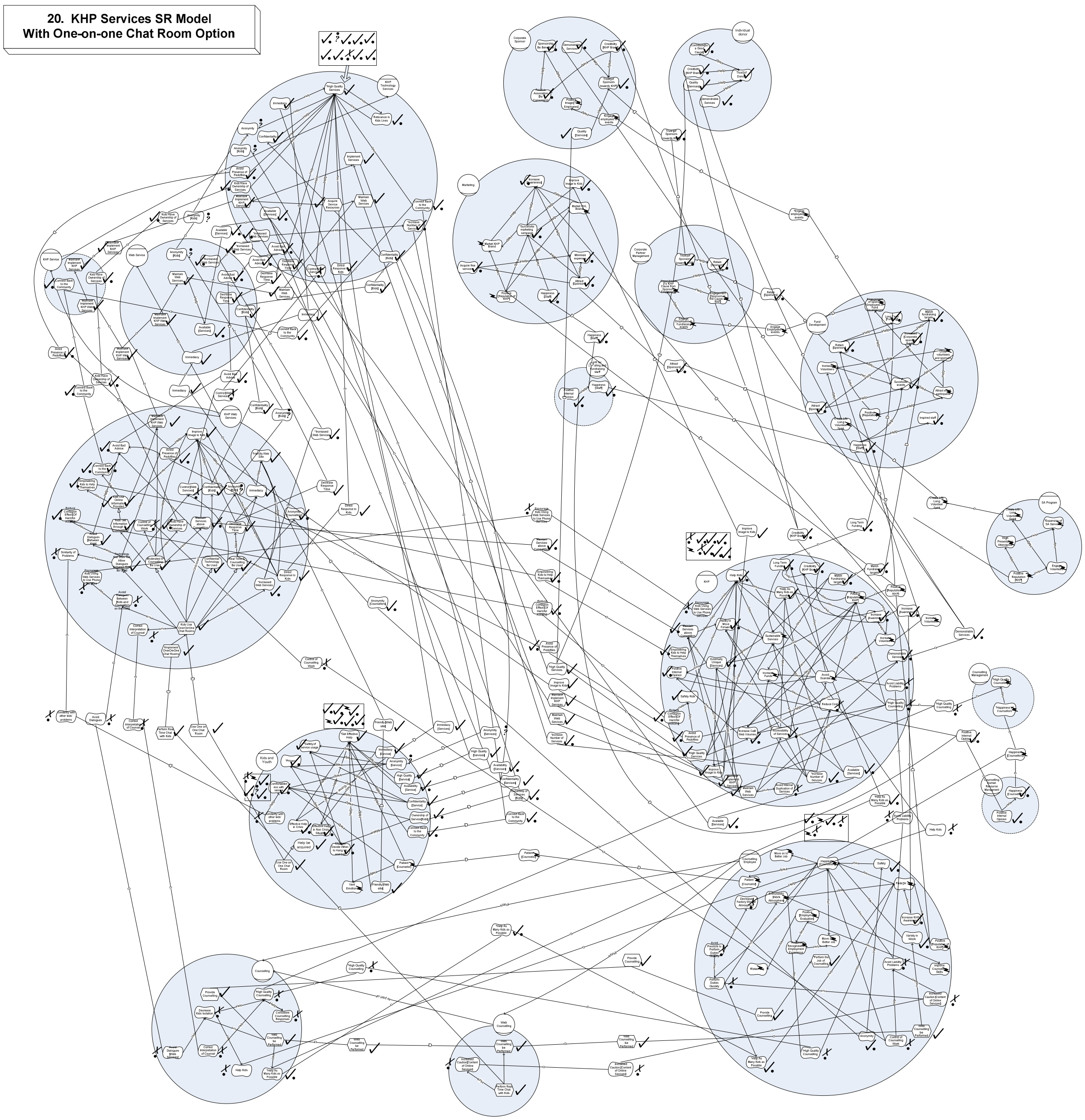


Figure 95 High-Level View of Counseling Service Model Analysis Alternative Evaluating One-on-one Chat Rooms

Lessons Learned (Stage 1). Although the process of modeling and analysis helped the analysts understand the organization and evaluate technology options, the models created were large and difficult to modify. As this was one of the first large, "real-life" agent-goal model case studies the investigators had undertaken, our initial approach was to model everything. Although our general mandate was to explore new technologies, we did not agree on a common scope for our analysis, and therefore tried to cover everything we learned about the organization in our models. The resulting models were difficult to understand, change, evaluate and were nearly impossible to validate with the stakeholders. We made compromises in the validation process by picking out smaller model excerpts representing key points in our findings, and presenting these to a group of stakeholders. These lessons lead of a focus on scoping as part of our suggested methodology for modeling and analysis (Chapter 8).

Forward analysis was used in this stage of the study to compare the effectiveness of technology options in large models of the organization. The procedure helped us to answer useful analysis question (which technique is more effective, and why?) (R13). However, procedure was difficult to apply manually on large models, leading to the need for at least partial automation (R3) and an implementation which hides procedure complexity (R16). As with the TC study, the analysis was able to accommodate high-level early RE concepts such as youth confidentiality, and quality of counseling (R9).

Stage 2. The next stage of the project focused on increasing the efficiency of the existing online counseling system. The evaluation procedure was used to analyze various configurations of a moderated bulletin board system. One large model focusing on online counseling was created (previously seen in Figure 86). Evaluation was applied extensively, but, as the models were created using Microsoft Visio, evaluation was done by hand. In this stage, the models were again too large to validate with stakeholders, so instead each option was presented, listing the important goals positively or negatively affected by the option in tabular form. The analysis results were stored in an excel spreadsheet, with softgoals and task options on the vertical, and analysis alternatives on the horizontal. High-level screenshots of these spreadsheets can be seen in Figure 119 and Figure 120 in Appendix E. In order to cope with the scale of analysis, results were converted to a quantitative scale, despite our arguments against the use of such scales in early RE. The quantitative results were used as part of a method to come up with priorities for potential features of the counseling system. These priorities were presented to stakeholders for verification along with the tables derived from parts of the complex model. The final outcome was a requirements specification document provided to the organization. Undergraduate students were employed to modify open-source software to meet the organization's specifications. Due to resource limitations and the risks involved in deploying the new system, the organization opted to modify their existing system.

Lessons Learned (Stage 2). Although the scope of this stage of the project was smaller than previous stages, the resulting model was still large and difficult to manage (in fact it was larger, as only one model was used). Manual evaluation over large models was time-consuming and error prone. The trade-off between completeness and model utility came to the forefront, emphasizing again the importance of scoping. This stage of the case study pushed the scalability and comprehension of the analysis procedure to its limits (and beyond), inspiring our description of the maximum reasonable model size in Section 11.4.1. We can conclude that for very large, detailed models, created during later stages of the system development process, fully automatic analysis could be more appropriate. Of course this raises issues concerning the trustworthiness and accuracy of analysis results, as addressed in Chapter 4, and discussed further in the future work section (13.3).

Stage 3. A later stage of the project focused on applying enterprise modeling to analyze the knowledge management needs of the organization. Models were again based on stakeholder interviews, but this time we created a first draft of each model "on-the-fly", with one of the analysts making a model of the interview content during the process of the interview. Models were later expanded, edited and reorganized based on interview transcripts and scoping decisions. Models were presented using "as-is" and "to-be" versions, including analysis, in order to show the effects of new technology options in the current situation. For example, Figure 96 shows an as-is analysis of the organizations referral database, while Figure 97 shows a to-be version with new technology options added to the model and analysis.

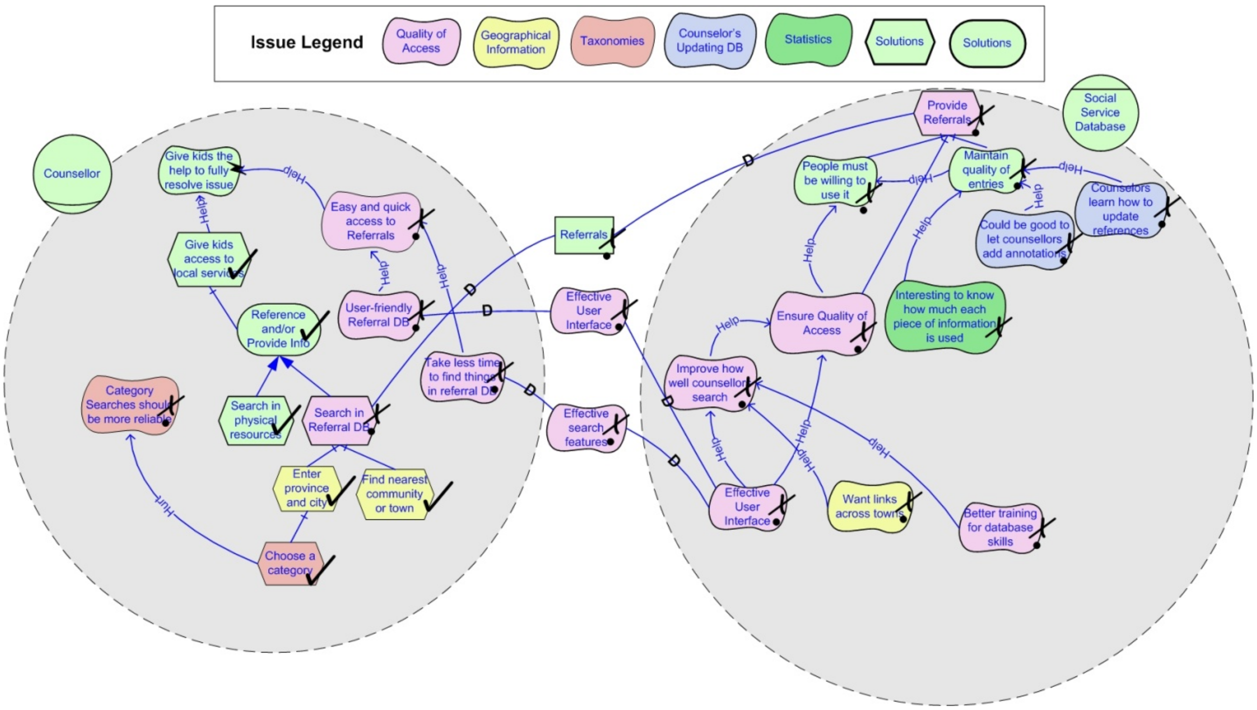


Figure 96: Referral Database As-is Model Showing Analysis Results

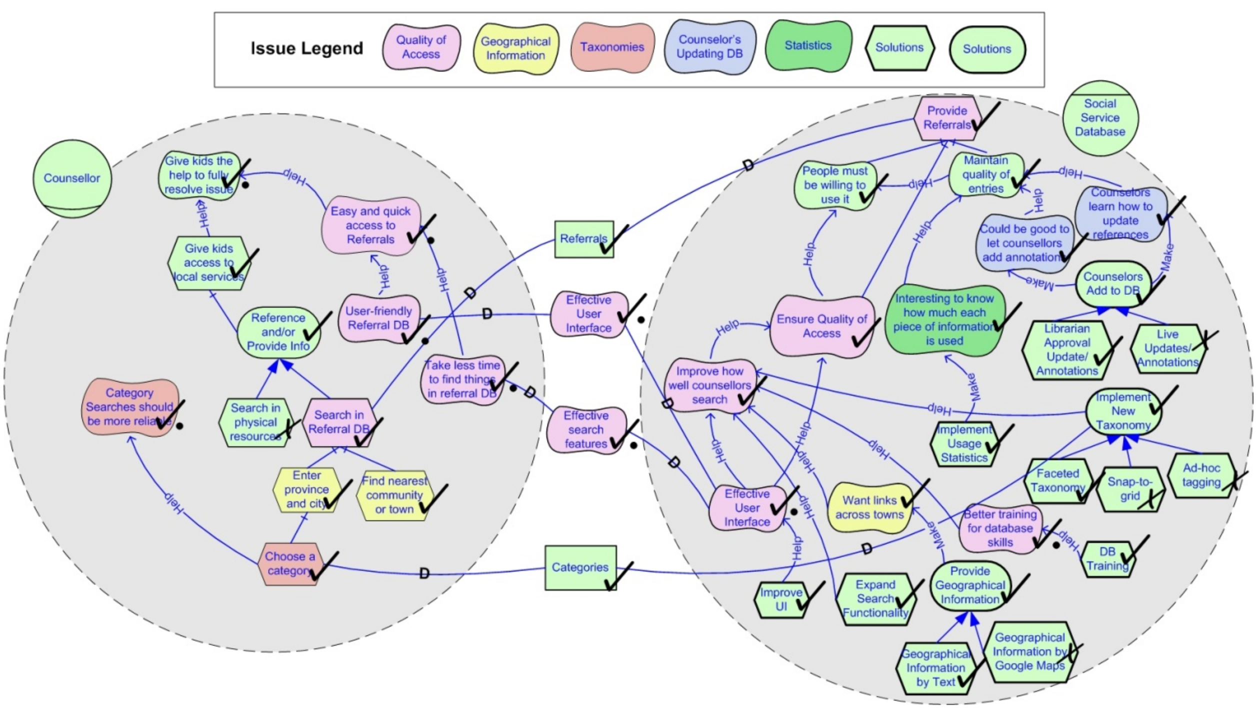


Figure 97: Referral Database To-Be Model Showing Analysis Results

The evaluation procedure was applied to several similar models in order to evaluate the situational effectiveness of a variety of technologies for storing and distributing knowledge, including wikis and discussions forums. It was discovered, for example, that the features of a wiki were not effective in satisfying the goals of the organization, while a discussion forum, with a set of specific features, showed more promise. See Figure 92 for a simplified model with analysis depicting the effectiveness of modified internal forums. We found the analysis procedure to be effective in facilitating a comparison between technologies, with the results reported back to the organization in reports and presentations, receiving positive stakeholder feedback.

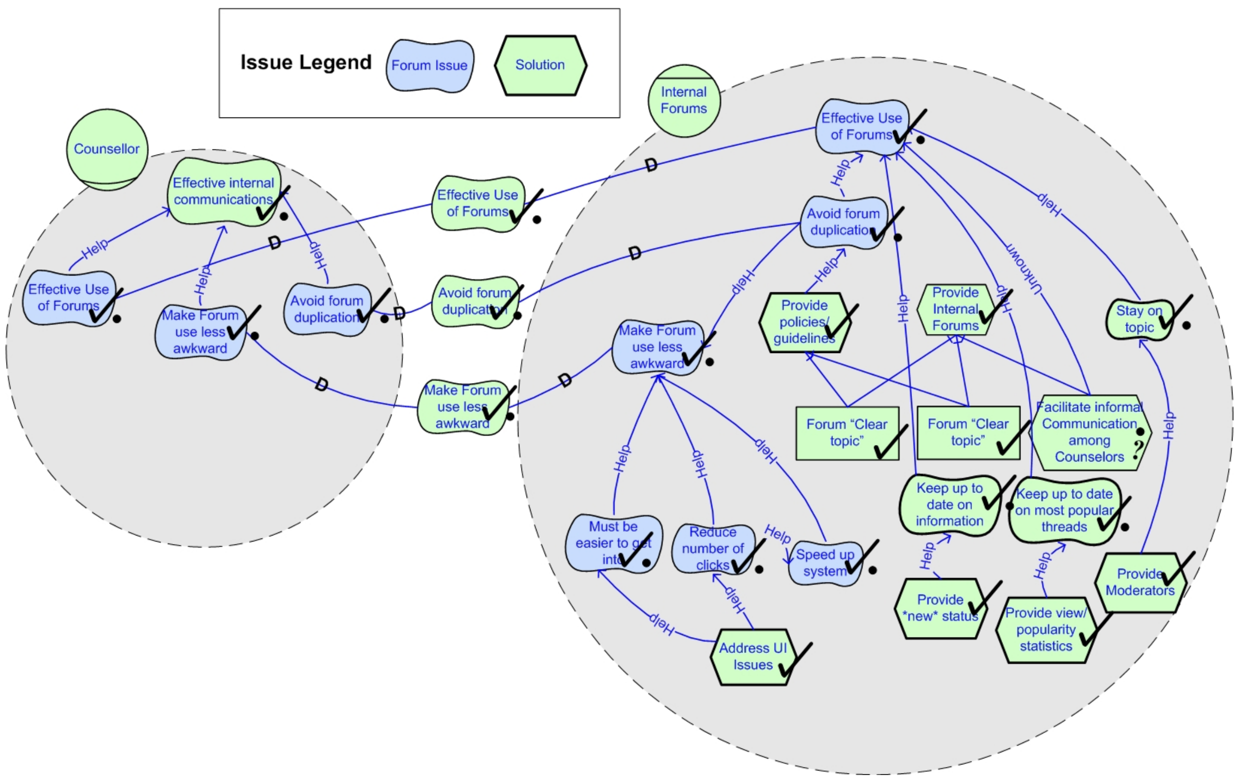


Figure 98 Forums model, to-be, evaluated

In the conduction of modeling and interactive analysis case studies up to this point, we had observed evidence of model iteration provoked by evaluation. In this stage we began to collect measures of such iteration. For example, a model focusing on communication contained 181 links and 166 elements before evaluation, while after evaluation the same model had 222 links and 178 elements, a difference of 41 and 12 respectively. In another example, the link count rose from 59 to 96 and the element count rose from 59 to 76. These numbers do not take into account changes such as moving links or changing element names. Models in this stage of the study were created by three individuals, with evaluation performed by two individuals, indicating that this effect is not specific to a particular modeler or evaluator.

Lessons Learned (Stage 3). Although the models in this stage were still large, we were much more rigorous with scoping decisions. Each model focused on one specific issue at the core of the organization's Knowledge Management issues. As a result, the models were easier to understand, modify and evaluate. Creating the models "on-the-fly" made the modeling process easier for the analysts; however, the models were not created in such a way as the stakeholders could view or participate in the modeling process, so this technique did not assist in model validation. Simplified versions of the models and analysis were shown to stakeholders in presentations, with positive feedback. We provided the models and analysis results to the organization, supplemented by textual description. However, it is difficult to know if the stakeholders would have been able to read and get value out of the models and analysis without our guidance.

The analysis procedure was again helpful in answering analysis questions (R13) in this case, analyzing the situational effectiveness of knowledge transfer technologies. The focus on scope produced models over which analysis was more easily applied and understood, helping to demonstrate the scalability (R1) and comprehension (R2) of analysis over models of a reasonable size.

Pattern Study Using Forward Analysis. Work described in (Strohmaier et al., 2008) explored the application of patterns for the construction of i* models. The work uses two case studies or experiments in order to test their hypothesis concerning patterns: stage 3 of the counseling service study, focusing on knowledge management technologies, and a small experiment with students in a system analysis class. The work hypothesizes that the use of patterns may produce models which are more complete and over which modelers have increased confidence. In the case study, the patterns were used to represent the promising knowledge management technologies such as a wiki or discussion forum. See Figure 99 for an example i* pattern of a wiki, showing the results of forward qualitative analysis.

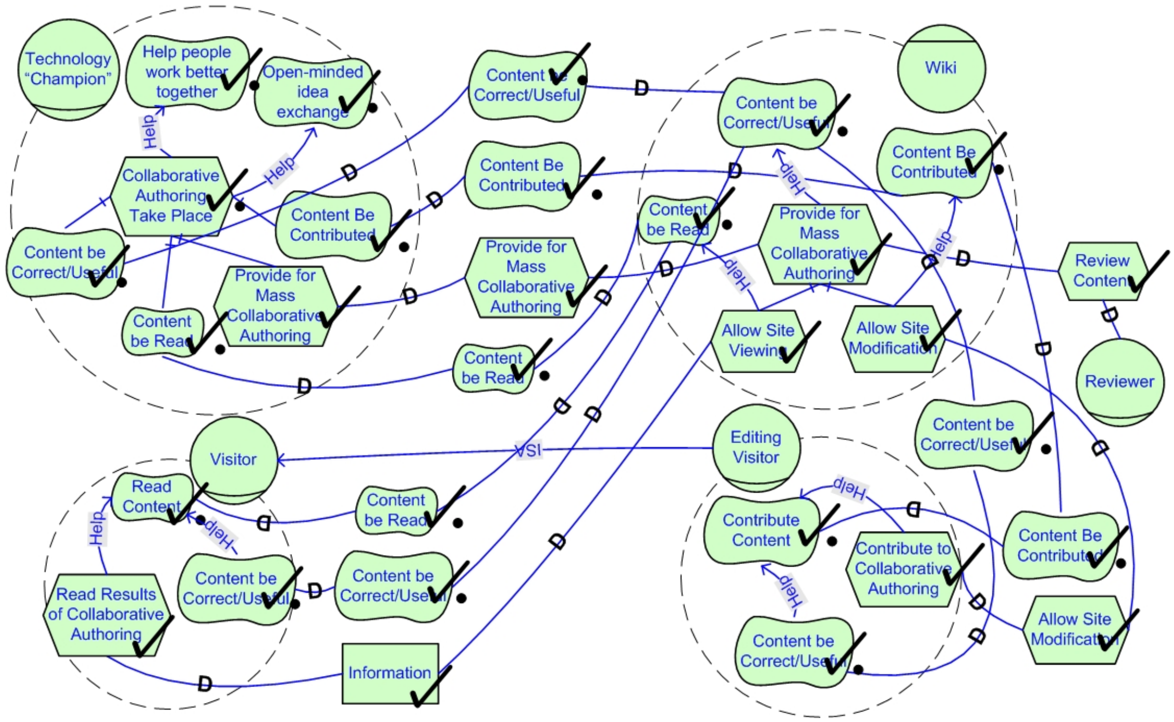


Figure 99 Example of an i* Pattern depicting a Wiki and the Results of Forward Analysis from (Strohmaier et al., 2008)

The derived pattern application methodology involved forward qualitative analysis as described in this thesis in several stages. During pattern construction, forward analysis was recommended as a means to ensure that the goals of the pattern were achievable. Contextualized models representing the organization without the application of the new technology were evaluated (forward analysis) to obtain an as-is evaluation result for comparison. Once a pattern had been integrated into the contextualized model, the entire model was re-evaluated to compare the achievement of goals from as-is to the potential to-be situation, evaluating the effectiveness of the technology represented by the pattern. See Figure 100 for a graphical depiction of this process. See Figure 101 for an example of the wiki pattern from integrated into a contextual counseling service model, showing the results of forward analysis.

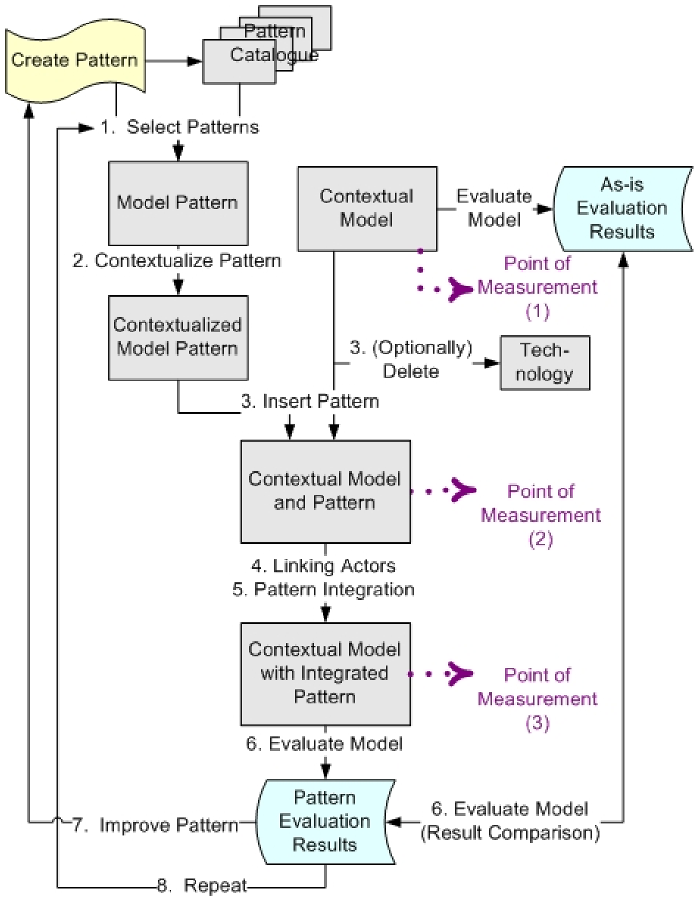


Figure 100 Pattern Methodology using Forward Analysis (Evaluate Model) from (Strohmaier et al., 2008)

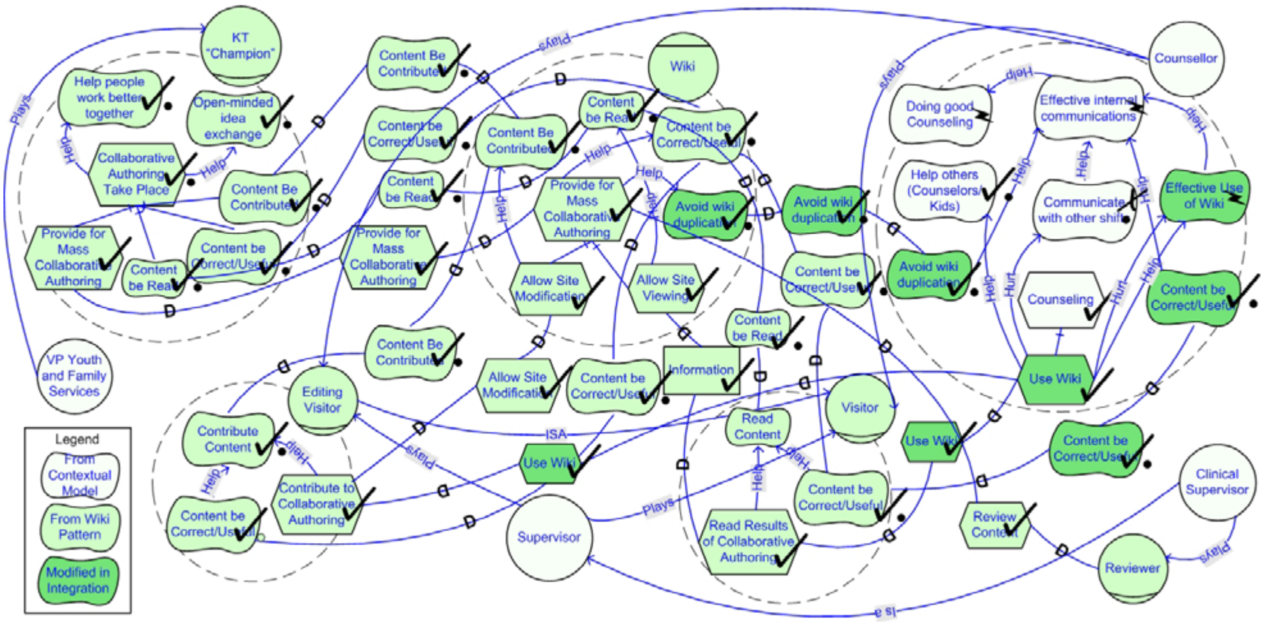


Figure 101 Contextualized Model with Integrated (Figure 99) Pattern Example (Evaluate Model) Showing Forward Analysis Results from (Strohmaier et al., 2008)

The general findings of the study were that although the application of patterns actually made the modeling process more difficult and time consuming, the resulting models had more details and the modelers had more confidence in their correctness and completeness. This reveals that patterns have potential to improve the quality of goal models during their construction. The integration of the pattern methodology with the methodology suggested in Chapter 8 is left for future work.

Analyzing Knowledge Transfer Effectiveness. The counseling service case study was used to inspire a further technique, analyzing the effectiveness of knowledge transfer agents (Strohmaier, E. Yu, Horkoff, Aranda, & Easterbrook, 2007). This approach argued that the success of approaches for knowledge transfer in an organization relied as much on organizational context, including goals and dependencies than on the technical details of the transfer mechanism. The proposal was to combine existing ideas concerning knowledge transfer with agent-oriented models, describing a technique which guided modeling and analysis of knowledge transfer "agents". The approach used the experience factory example, and further, detailed examples from the counseling service study. Forward analysis as described in this work was used to analyze the effectiveness of the new knowledge transfer agent. For example, Figure 102 shows an analysis of whether or not a discussion forum knowledge transfer agent can satisfy the goals of a knowledge consumer and provider, given the analysis alternative with no notification or moderation.

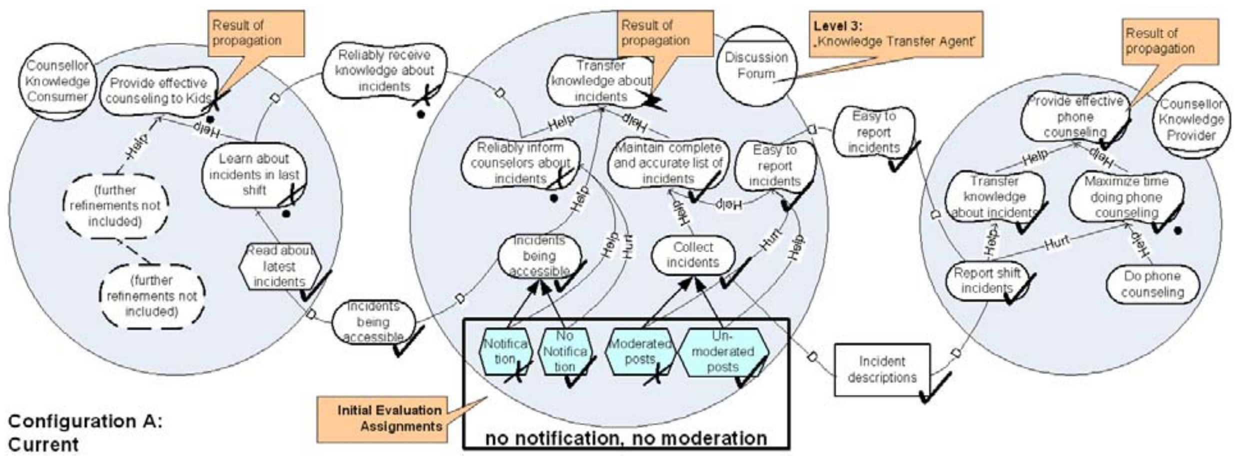


Figure 102 Example Knowledge Transfer Agent Analysis for a discussion Forum (Model created by M. Strohmaier)

The introduction and use of this method on small and medium examples helped to further demonstrate the comprehension (R2), analysis capability (R13) and simplicity (R15) of the forward analysis procedure.

12.1.4.3 Study Contributions

Although the process of modeling and analysis helped the analysts understand the organization and evaluate technology options, the initial stage models were large and difficult to modify. Our initial approach was to model everything, without a careful consideration of scoping decisions. In later stages, we were much more rigorous with scoping decisions. Each model focused on one specific issue at the core of the organization's Knowledge Management issues. As a result, the models were easier to understand, modify and evaluate. The forward analysis procedure was shown to be scalable (R1) and comprehensible (R2) to several users over models of a reasonable size.

Overall, we found that drawing and analyzing i* models demonstrated the ability of the approach to aid in domain understanding, analysis, decision-making, and communication. The procedure allowed effective comparison to technology options in the domain (R13). The evaluation procedure especially helped to provoke changes in the model (R4) which improved the quality of the model and forced the modeler to learn more about the domain.

Ideally, future studies would involve the stakeholder directly in the modeling and analysis process. Prompted by our case study experience, an exploratory experiment was developed and carried out an in order to test some of the perceived benefits of the procedure.

12.1.5 Exploratory Experiment

Observations in case studies have shown that the evaluation procedure described in this work aids in finding non-obvious answers to analysis questions (R13), and prompts iteration over the model (R4). We have seen experiential evidence that such effects lead to further elicitation, and lead to a better understanding of the domain. Our experiment begins to test whether these effects are specific to our procedure or are a product of any detailed examination of a model. We are also interested in how modeling and evaluation experience as well as an evaluators' role in creating a model affect our results.

12.1.5.1 Design

The experiment materials were taken from a study applying goal-oriented analysis to the sustainability issues for the ICSE conference (Cabot et al., 2009). The study produced a series of models focusing on various actors in the domain of conference planning, focusing on the tradeoffs between greening and non-greening goals for the conference chairs. The participants of this study, including one of the authors, having knowledge of the domain, were asked to participate in a further study testing the effects of the procedure described in this work. The participants were asked to evaluate two different questions over three models, once without using the procedure and, after training, once using the forward, interactive procedure described in this work. The results were compared in terms of analysis findings, questions discovered, model changes, and time taken.

Models and Questions. Three models (M1, M2, and M3), each created by one or more participants, were selected for use in this experiment. M1 focused on the Publicity and Program Committee Chairs and contained 55 elements, 82 links and 8 actors (Figure 55); M2 focused on the Conference Experience Chair and contained 36 elements, 50 links and 5 actors; while M3, containing 78 elements, 130 links and 15 actors, focused on the General Chair (Figure 40).

Participants were given two different questions (Q1 and Q2) specific to each of the three models. Questions were developed to be non-trivial, and either explored the effects of a particular set of options on high-level goals or asked more general questions related to the possibility of goal satisfaction. For example, Q1 over M1 "If the Publicity Chair distributes materials online and the PC Chair prepares only online proceedings and has only online submissions, how will this affect the significant goals of the actors (acceptance rate, quality of program, diffusion, etc.)?" and Q1 for M2 "If every task of the Sustainability Chair and Local Chair is performed, will goals related to sustainability be sufficiently satisfied?".

Participants. The participants were all current or former researchers in Computer Science or Information Systems disciplines. Three of the participants (P1, P2, and P4) were new to i* before creating the models in question, another participant, P3, had extensive experience with goal models, and one participant (the thesis author, P5) could be considered an "i* expert". P1 to P4 were not aware of the specific hypothesis of this study.

Experimental Steps. Due to the small number of participants, we did not split the participants into groups using, and not using, the procedure. Instead participants evaluated models not using, and then using the procedure, examining the changes and additions between results. In this set-up participants would already be familiar with the analysis results before applying the procedure. Therefore, the focus will be on examining the effects of the procedure beyond what can be gained by ad-hoc analysis.

For each of the three models, participants were asked to describe their role in creating the model, record their answers to the analysis questions, and record any question derived from their analysis. They were then asked to familiarize themselves with the evaluation procedure by reading a manual, now available online, containing an explanation of the interactive evaluation procedure (Horkoff, 2009). Participants then re-evaluated each model using the procedure, again capturing questions, model changes and analyzing the differences between their analysis results. They were asked to record the time taken for each step.

After all steps, participants were then asked to answer several follow-up questions: Did model changes improve the quality of the model? Do you have a better understanding of the model and domain? Did this increase more or less, with or without using the procedure? Would you use the procedure again?

12.1.5.2 Results and Analysis

We examine several aspects of the results. First the differences in analysis results not using, and then using the procedure, helping to show that the procedure finds non-obvious analysis answers (R13). Table 63 shows each of the areas of change and how many participants made a particular type of change in that area. Here, we capture whether or not specific types of changes were made to the analysis results by participants after applying the procedure. The results are a count of participants and not a count of specific changes. For each model, there are two questions over which results could change with application of the procedure, making six possible areas of change. For example, NP_Q1 to P_Q1 for M1 explores the differences between the answers for Q1 concerning M1 with no procedure (NP) and the procedure (P). For each area, we categorized changes under one or more categories: no change, one or more changes in strength (partial to full satisfaction/denial or vice versa), one or more changes in polarity (a change between one of partial/full satisfied, partial/full denied, and conflict), more intentions evaluated (more intentions included in evaluation results), and less intentions evaluated (intentions which were included in evaluation results not included). For example, in applying the evaluation procedure to Q1 for M2, one participant made no change from their previous results, three participants made changes in strength, and one participant evaluated fewer intentions.

In Table 64 we provide summary data which shows the changes by participant, summing over all models and questions.

Table 63 Analysis Result Changes – Number of Participants Making Changes of each Type for each Question and Model

	M1		M2		M3		
	NP_Q1 to P_Q1	NP_Q2 to P_Q2	NP_Q1 to P_Q1	NP_Q2 to P_Q2	NP_Q1 to P_Q1	NP_Q2 to P_Q2	Sum
No Change	1	0	1	1	3	0	6
Change in strength	2	1	3	2	0	2	10
Change in polarity	3	1	0	3	1	2	10
More intentions evaluated	3	3	0	0	1	2	9
Less intentions evaluated	1	2	1	0	0	1	5
Sum	10	7	5	6	5	7	40

Table 64 Analysis Result Changes – Number of Changes per Type for Each Participant over All Models and Questions

	P1	P2	P3	P4	P5	Sum
No Change	3	1	0	1	1	6
Change in strength	2	0	3	3	2	10
Change in polarity	0	2	2	4	2	10
More intentions evaluated	1	1	2	2	3	9
Less intentions evaluated	0	3	1	1	0	5
Sum	6	7	8	11	8	40

Generally we can see that participants did make changes in their analysis results when applying the evaluation procedure. When using a more consistent way to propagate evidence, users change label strength, change polarity, notice new paths of intentions missed, and notice paths which were not actually affected by the alternative. These observations help to demonstrate that the procedure helps to answer analysis questions beyond ad-hoc analysis (R13).

Although we see that changes are made, we must question whether these changes produce more accurate results. One participant found the evaluation procedure to be too conservative, marking intentions as partially satisfied/conflict that were previously judged to be satisfied/partially satisfied. In such cases, we would hope that the evaluator would use this as a catalyst to modify the model, but in this particular case, the only changes made were before the evaluation procedure was applied. However, the same participant stated for a different model that: "…the evaluation showed the model's weaknesses more clearly." Overall, results seem to reveal inconsistencies between the model and the user's perception of the domain.

Next, we count the changes made to the models not using and using the procedure (R4). We classify changes to models in several categories, counting results on an individual basis. Table 65 shows the count of each type of change made for each model during each question analysis, where the numbers are summed over all participants. In addition to the changes listed in the left column, we also measured changes to element names, adding an actor, and changing an actor name; however, none of those changes were made by the participants in this study. Results showing the breakdown per participant, summed over models and questions are shown in Table 66. We can see that all participants made changes, with the total number of per participant changes ranging from 8 to 35.

Table 65 Model Changes using and not using Systematic Analysis, Summed over Participants

	M1				**M2**				**M3**				
Categories	NP_ Q1	NP_ Q2	P_ Q1	P_ Q2	NP_ Q1	NP_ Q2	P_ Q1	P_ Q2	NP_ Q1	NP_ Q2	P_ Q1	P_ Q2	Sum
Add Link	0	0	1	0	5	4	0	2	2	6	0	2	22
Remove Link	1	0	0	0	5	7	3	0	0	9	0	3	28
Add Intention	0	0	1	0	1	2	0	0	0	1	0	1	6
Remove Intentio	1	0	0	0	1	2	2	0	0	3	0	3	12
Change Link	4	3	1	2	0	0	2	4	1	0	0	0	17
Change Intent Type	1	0	0	0	1	0	1	0	0	0	0	0	3
Remove Actor	0	0	0	0	0	1	0	0	0	0	0	0	1
Move Link	2	0	2	1	2	3	4	4	0	2	0	1	21
Move Intention Actor	0	0	0	0	1	0	0	0	0	0	0	0	1
Sum	9	3	5	3	16	19	12	10	3	21	0	10	111

Table 66 Model Changes for each Participant using and not using Systematic Analysis, Summed over Models and Questions

	P1		P2		P3		P4		P5		
Categories	NP	P	NP	P	NP	P	NP	P	NP	P	**Sum**
Add Link	0	0	1	0	5	4	0	2	2	6	22
Remove Link	1	0	0	0	5	7	3	0	0	9	28
Add Intention	0	0	1	0	1	2	0	0	0	1	6
Remove Intention	1	0	0	0	1	2	2	0	0	3	12
Change Link	4	3	1	2	0	0	2	4	1	0	17
Change Intention Type	1	0	0	0	1	0	1	0	0	0	3
Remove Actor	0	0	0	0	0	1	0	0	0	0	1
Move Link	2	0	2	1	2	3	4	4	0	2	21
Move Intention in Actor	0	0	0	0	1	0	0	0	0	0	1
Sum	9	3	5	3	16	19	12	10	3	21	111

Results show that more changes were made during the initial analysis without the procedure (71 changes) than with the procedure (40 changes). These columns are shaded with blue. These results are somewhat surprising, indicating that the iteration provoked by the procedure may have more to do with forcing the user to carefully manually examine the model than with the procedure itself. This leads to a further hypothesis left for future investigating: more automated procedures would be less likely to provoke model iteration.

We note that the participants found 40 additional changes using the procedure to answer the questions for the second time, future studies should make use of two participant groups in order to measure if second round ad-hoc analysis would also produce additional changes. Although we can see that changes were made, we can also examine whether or not the changes were perceived as beneficial. Three out of five participants said that changes made to the models improved the quality of the model. These participants indicated the quality was improved through changes made both with and without the procedure. The other two participants did not feel they had made significant changes to the models in either stage, with one stating that “additional knowledge information would be needed to really improve the quality of the models”, and the other echoing the sentiment. These results help to emphasize the incomplete and iterative nature of such models, and their ability to prompt further elicitation.

To further investigate the ability of the procedure to prompt elicitation, potentially increasing the completeness and accuracy of the models, we look at the number of questions the participants came up with when finding answers to the models. The questions collected are categorized in three ways: a specific question concerning the domain, a question which points out a flaw in the model, and a general comment concerning the need to expand the model. Table 67 shows the results summed over all participants. As Table 67 shows, all participants came up with various types of questions, with the number of questions per participant ranging from 5 to 16. We can see that many of the "questions" were actually participants pointing out flaws in the model, providing further evidence supporting the ability of the procedure to provoke model changes.

Table 67 Questions Found Summed over Participants

	M1				M2				M3				
	NP_Q1	NP_Q2	P_Q1	P_Q2	NP_Q1	NP_Q2	P_Q1	P_Q2	NP_Q1	NP_Q2	P_Q1	P_Q2	Sum
Specific Question	4	1	4	5	0	3	1	1	1	0	1	1	22
Flaw in Model	2	5	2	0	2	2	0	1	2	2		1	19
Need for expansion	0	0	0	0	1	1	2	0	0	0	0	0	4
Sum	6	6	6	5	3	6	3	2	3	2	1	2	45

Table 68 Questions Found per Participant, Summed over Model and Questions

	P1		P2		P3		P4		P5		
Categories	NP	P	NP	P	NP	P	NP	P	NP	P	Sum
Specific Question	2	0	0	1	1	0	2	10	4	2	22
Flaw in Model	3	1	4	2	4	0	3	1	1	0	19
Need for expansion	0	1	2	1	0	0	0	0	0	0	4
Sum	5	2	6	4	5	0	5	11	5	2	45

The results in Table 67 and Table 68 show that 26 questions were derived without using the procedure, with an additional 19 derived using the procedure. These results are again interesting in that they show careful examination of a model through ad-hoc analysis leads to further elicitation. We could hypothesize again that this effect may not occur with automated analysis. It is promising to note that even though many questions were found without the procedure, application of the procedure provoked a significant number of further questions, even though the same analysis questions were being evaluated. Future studies need to test whether a second round ad-hoc analysis would produce the same results as a second round of systematic analysis did in this case.

All five participants reported a better understanding of the domain after this exercise, with all participants claiming that they gained a better understanding using the evaluation procedure than using no procedure. Specific comments include:

> "The procedures helped to identify where there were conflicts (which often indicated problems in the models(more) than a true conflict in the situation), which I did not see just by evaluating "intuitively". When the evaluation procedures resulted in "undecided" labels, it emphasized the problem (or lack of information) in the analysis questions themselves rather than in the models."

> "Your automated procedure was overall more helpful mostly because it kind-of offered me somebody to argue with about my own intuitions."

Table 69 reports the time taken by each participant to perform each analysis question, including a column for training time

Table 69 Time Taken by each Participant for Each Analysis Question

	M1				M2				M3				
Participant	NP_Q1	NP_Q2	P_Q1	P_Q2	NP_Q1	NP_Q2	P_Q1	P_Q2	NP_Q1	NP_Q2	P_Q1	P_Q2	**Training**
P1	10	15	5	10	10	5	5	5	5	10	2	10	15
P2	10	15	30	25	15	10	5	10	20	10	10	15	30
P3	6	7	10	10	7	5	15	20	5	20	5	15	40
P4	10	12	15	10	8	7	13	8	12	16	11	12	40
P5	5	3	10	15	3	10	10	7	9	5	5	10	0
Average	8.2	10.4	14	14	8.6	7.4	9.6	10	10.2	12.2	6.6	12.4	31.25

The average time to answer a question without the procedure was 9.5 minutes (standard deviation of 4.6) compared to 11.1 minutes (standard deviation of 6.0) using the procedure. Although the variance is high, we see that working with procedure takes only slightly more time than without. In terms of ownership, P1 had a role in creating M1, P2 and P4 had a role in creating M2, and P5 had a role in creating M3. Results do not clearly indicate if the role in creation affects the results measured. The same holds for experience, P5 (the author) who had the most experience with the procedure, did not produce results which stood out significantly from the other participants. Finally, all five participants said they would use the procedure again if they had to evaluate another i* model.

12.1.5.3 Study Contributions

Results of our exploratory experiment indicate that the evaluation procedure prompts changes to analysis results and may prompt model iteration and elicitation beyond analysis without a systematic procedure (R4). The participants have reported that the procedure provides a better understanding of the model and domain. The experimental design has implicitly tested the simplicity of the procedure (R15) as the participants were able to learn and apply the procedure manually and the scalability of the procedure over medium-sized models (R1). However, the experiment suffers from several threats to validity, including the small number of participants.

Using the lessons learned from this experiment, we conduct further experiments with more participants in the next sections. Thus far, we have only investigated model evaluation in the context of a single modeler. The implementation of the procedure in OpenOME better enables further case study application.

12.2 Application of Forward and Backward Analysis Using the OpenOME Implementation

We have assumed in this work that more utility can be gained from goal models by applying explicit analysis over models, but we have yet to understand how or why this occurs. In this work we test hypotheses concerning interactive goal model analysis via multiple case studies as described in the introduction to this chapter. Previous case study results have indicated that such analysis increases model iteration, prompts further elicitation, and improves domain knowledge. Results of the next round of studies do not provide strong evidence to support these claims, showing that such benefits, when they occur, can occur both with systematic and ad-hoc model analysis. However, the results reveal other benefits of systematic analysis, such as a more consistent interpretation of the model, more complete analysis, and the importance of training.

In this work, we designed and administered two types of case studies to further test the hypothesis concerning interactive analysis suggested by previous work. Following our earlier work, we use i* as the goal modeling framework in these studies. Due to the great number of confounding variables, we chose to use case studies as the research method, rather than experiments producing statistically significant data. Specifically, we conducted ten case studies using subjects with some experience in i* modeling. Half of the participants analyzed models using no explicit procedure (ad-hoc analysis) while the other half used implementations of the forward and backward interactive analysis procedures.

We have previously hypothesized that interactive analysis provokes useful group discussions. In order to gain some insight into analysis by individuals versus analysis in a group, we administered a separate multi-session case study involving a project team designing tool support for modeling "back of the envelope" calculations.

Qualitative and quantitative analysis of results are used to compare treatments in both studies, to gather evidence to support or deny our claims concerning the benefits of the analysis procedures, and to gain an understanding of further benefits of and barriers to systematic goal model analysis, helping to guide the application of goal analysis for systems within an organization.

12.2.1 Case Study Design

We designed and administered ten case studies involving individuals and one multi-session case study with a group of participants, all applying interactive analysis over i* models. In the first type of study, our unit of analysis was the individual participants, while in the second it was the group as a whole. As our aim was for interesting qualitative and quantitative findings without statistical significance, changes were made to the procedure under analysis and to the case study designs at various points. We describe the initial and modified study designs in the following. Study design choices and threats to validity are discussed in Section 12.2.6.

12.2.1.1 Individual Case Studies

Overview. The studies were administered in two rounds. In the first round, six participants were provided i* refresher training and instructions for the study. They were given an introductory sheet describing the model domain, introduced to the three subject models and twelve analysis questions, then given time to answer the questions over the models. In the second round, four participants were given i* refresher training and study instructions, then spent about 25 minutes creating an i* model about life as a student, and then followed an analysis methodology which guided the application of various questions over the model. In both rounds, half of the subjects used the systematic analysis procedure while the other half answered the questions using ad-hoc analysis. The subjects using systematic i* analysis received an additional round of training for the forward and backward procedures (15 minutes). All participants were told that they could make changes to any model at any point, but that they should not feel obligated to do so. The study involved a "think-aloud" protocol, with the thesis author present to observe the progress and answer questions. Participants were encouraged to ask questions about the model if they had them. Results were recorded via audio recording, screen capture and saving versions of the models. All participants used the i* drawing implementation in the OpenOME tool. Every participant was asked a series of follow-up questions concerning their experience. The total time for each study in both rounds was two hours or less.

Participants. Participants were recruited via a call for participation to students who had learned about i* in one or more system analysis courses, or to students involved in i*-related tool or research projects. Selection was purposive rather than random, we wanted subjects with some knowledge of i* but who were not very familiar with goal model analysis of any form. The resulting participants were students at either the graduate or undergraduate level in Computer Science, Information Systems or Health Informatics. The students had previously created anywhere from one to ten i* models of varying detail, all within the last year. Participants had from none to ten years of industry experience, mostly in technical-related fields. Subjects were paid $40 regardless of the time taken or the results, and results were not made available to anyone who had an influence on course evaluation.

Training. The first two participants of Round 1 were given an i* refresher handout, reminder the participant about i* concepts and analysis labels. The subject using the systematic analysis procedure was given a similar handout describing the forward and backward procedures. After these initial runs of the study, it was apparent that the subject's i* knowledge was not particularly strong. The time devoted to reading the refresher and training documents was not significant. The study was revised such that the facilitator gave a ten-minute i* refresher lesson, and for the participants using systematic analysis, a 10-15 minute instruction session describing the analysis.

Model Domain. In Round 1, subjects were asked to analyze models from the ICSE Greening domain, the same domain used in the exploratory experiment in Section 12.1.5. Three models were used from this study, containing between 36 and 79 intentions, 50 and 130 links, and 5 and 15 actors.

The results of the first round of the study performed with six participants showed minimal model changes or elicitation questions, as well as participant difficulties in understanding the models. The decision was made to revise the study and instead allow participants to make their own models over a domain they were familiar with – student life. In the second round, the four participants were provided with some leading questions, (e.g., Who is involved? What do the actors want to achieve?), and then spent 25 minutes creating a model describing their student experiences. P1 to P6 used the ICSE Greening Models, while P7 to P10 used their own student models.

Analysis Questions. In the first round of study, twelve analysis questions over the three models were presented to the participants, four per model, two each aimed at forward and backward analysis. The questions were aimed to represent interesting questions over the domain. For example "If every task of the Sustainability Chair and Local Chair is performed, will goals related to sustainability be sufficiently satisfied?" (forward question) and "Is it possible for both sustainability and successful conference to both be at least partially satisfied? If so, how?" (backward question). The full list of questions can be found in Table 74 in Appendix E.

Results from the Group Case study, described in the next section, indicated that it was challenging to motivate modelers to analyze their own models, and that it was sometimes difficult for modelers to come up with interesting analysis questions. As a result, a suggested methodology for model analysis was created using our experiences in evaluating our models in practice. This methodology is described in Chapter 8.

As described, the first two sections were meant to act as "sanity checks" in the model, checking that it produced sensible answers for a variety of questions, while the second part was intended to support more useful analysis in the domain. Round 2 participants were asked to use this methodology to analyze the student life model they had created. The same methodology was used for all participants, as the version provided to participants did not explicitly reference the forward or backward analysis implementations.

12.2.1.2 Group Case Study

A second study was conducted involving a group of four graduate students and a professor who were in the process of designing and implementing a tool (Inflo) to support modeling and discussion of "back of the envelope" calculations. The participants wanted their tool to support informed debate over subjects, such as carbon footprint calculations, containing references to easily understandable models which themselves contain clear references to information sources.

Three two-hour modeling and analysis sessions occurred. Each session had one of the authors present as an i* expert and modeler, and anywhere from two to four of the participant group members. Most of the time in these sessions was devoted to constructing and discussing a large i* model representing the tool, its users, and their goals. During each session, some time was devoted to applying both the forward and backward analysis procedures, letting the participants make decisions over the human judgments posed by the procedures. In this study, the author/facilitator played more of a participatory role, drawing the model and administering the analysis with constant feedback and input from the participants. The first session concluded with a survey concerning the participant's experience with the analysis procedures, while the second and third sessions had audio and/or video recording. The large model resulting from this study has been used as an example in Chapter 9, demonstrating analysis visualizations.

12.2.2 Data Analysis Methods

The studies produced approximately 24 hours of audio and video, many versions of models, and pages of observer notes. Quantitative data was collected by counting how many and what type of changes to the model were made, (e.g., change a link type, add an intention), and how many domain-related questions were asked for each type of question for each participant (e.g., "What do they mean by collaborate?").

Qualitative data was coded as per the study hypothesis described in the introduction to this chapter, allowing for extra fields to capture additional interesting observations. The process of finding results not related to our initial hypotheses was similar to Grounded Theory (Seaman, 1999), where qualitative data was grouped according to relevant categories or codes relating to potentially interesting observations or theories. Analysis of further subjects potentially added more evidence to these categories, or produced new categories. What resulted was a list of interesting observations or theories with an associated list of qualitative support classified by participants.

12.2.3 Study Results

Analysis aids in finding non-obvious answers to domain analysis questions. Results for this hypothesis were mixed. Some participants gave explicit answers to the questions, some referred to analysis labels in the model as answers to the question, while yet others had difficultly producing answers to the questions. One participant was not sure when they were done answering a question. Ideally, participants would be able to interpret the results of the question in the model in the context of the domain; however, only some participants were able to do so. Similarly, participants often had difficulty in translating questions into initial labels in the model.

These results point to a difficulty in mapping the model to the domain, both in starting the analysis and in translating the results back to the world. Presumably, this is a skill which comes with modeling experience. It is interesting to note that these difficulties seemed more prevalent in Round 1 where participants were analyzing large models created by others. It seems that knowledge of i* and the domain may have a significant effect on the ability to apply and interpret analysis.

Results for this question for the inflo group study were difficult to interpret. The modelers themselves did not have any driving domain questions; therefore the analysis questions asked were somewhat artificial. Some analysis alternatives helped to test the sanity of the model by answering domain questions, for example, if the inflo system was built, the trolls (malicious inflo users) win, according to the model.

Model Iteration: prompts improvements in the model. Counts of the number of changes made for each participant are shown in the 3rd and 4th columns of Table 70. Generally, few changes were made with the exception of P1 who redrew much of the model at the start of the study independent of the analysis questions. We omit detailed data on types of changes; however, some specific examples include changing decomposition to contribution links and adding or renaming tasks. Note that a few changes were suggested by the participants but not made, and are not included in the counts. The number of changes was not significant for most participants, and there are more changes made with ad-hoc than systematic analysis. There is no notable difference between participants analyzing their own or others models. For the five participants who made changes, we asked if those changes were helpful, four said yes, while one said it depends on whether changes would be helpful to domain experts.

We see no significant differences between results for round 1 and 2, where round two were given a more in depth analysis methodology, as described in Chapter 8.

Table 70 Number of Model Change and Questions Asked for each Participant

		# Model Changes		# Questions Asked		
Treatment	Participant	Forward Questions	Backward Questions	Forward Questions	Backward Questions	Round
	P1	59	10	10	1	
	P4	0	0	1	0	
	P5	5	13	6	6	1
	P7	2	5	0	0	
Ad-hoc	P9	0	5	0	0	2
	P2	0	0	2	3	
	P3	0	0	2	0	
	P6	0	3	5	1	1
	P8	0	0	2	2	
Systematic	P10	0	0	0	1	2

Analysis did prompt some changes in the inflo case, for example, removing links, but the changes were not extensive.

Elicitation: leads to further elicitation of information in the domain. The number of domain-related questions asked by each participant is shown in the 5th and 6th columns of Table 70. Again, we see no interesting differences between groups.

In the inflo case, the modelers and the stakeholders were the same group, so any questions raised by the modeling or analysis process were discussed and resolved immediately. We return to these findings later in this section.

Domain Knowledge: leads to a better understanding of the domain. At the end of every individual study, we asked: do you feel that you have a better understanding of the model and the domain after this exercise? Seven out of ten participants said yes. One participant who did not say yes was commenting on the complexity and learning curve associated with i*, another complained that they were already very familiar with being a student, and didn't learn anything further, and the last said that they learned more about the model, but not about being a student. Selection of complex models and a familiar domain seemed to hinder this potential benefit. Analysis was helpful for both systematic and ad-hoc approaches. Participants provided specific comments concerning evaluation: analysis brings out the flaws in the model, and it was helpful for understanding the effects of goals and relations and in choosing between alternatives.

Feedback through surveys for the inflo group revealed that analysis helped clarify tradeoffs, and the meanings of intentions, although several usability issues with the procedure were found.

Promote Discussion in Group Setting. Application of systematic evaluation in a group setting did produce several situations where human judgment caused discussion among participants. For example, the participants discussed whether getting feedback was really necessary in order to make models trustworthy after this contribution appeared in a backward judgment situation for Make models trustworthy. In other examples, the group had discussions about the exact meaning of goals appearing in judgments situations, for example "what is meant by Flexibility?" This revealed that different participants thought it meant slightly different things. To be fair, not all judgment situations provoked discussion; more experience is needed to determine how to maximize this positive effect.

12.2.4 Analysis

We can try to understand the results for model changes and elicitation, and why they differ from the results found in previous studies, by examining the reasoning behind these hypotheses. Previous studies have claimed that it is the interactive nature of the analysis that prompts for changes to the model and drives elicitation. We can expand on this claim, by considering the differences between a goal model representing a domain and the participant's mental model of the domain. An i* model can be considered an incomplete representation of the mental model of its creator. When human judgment is needed in a model, the evaluator is asked to use their mental model of the world to supplement the contents of the physical (explicitly expressed) model. The hypotheses rely on differences between the mental model of the participant and the explicit i* model, especially if they were not the creator of the model. Although such differences could be discovered at any point, they may become particularly apparent when answering human judgment questions.

When these differences are discovered, they may prompt changes to the model, or may cause inquiries concerning the domain. For an example of the former, in the Inflo case study, when asked "Is it possible to make (Inflo) models at least partially trustworthy?" one of the participants decided that validation of a model was not relevant to trustworthy, and the link was removed. The model did not match that participant's mental model of the domain. In other cases, missing elements, inaccurate contributions, or questions concerning the meaning of elements could arise. For example, a human judgment concerning Make conference participation fun made one participant make changes to the model to make the conference more fun and sustainable, renaming task and changing a link from a hurt to a help.

Because a small number of changes were made to the model, and a modest amount of questions were discovered, we can hypothesize that either the evaluation did not typically reveal differences between the mental model of the evaluator and the explicit model, or these differences existed, but were not used to modify the model. We can find several examples where the evaluator seemed to ignore the structure of the model and answer human judgment questions using only their mental model. For example, in one case, in the forward judgment for the softgoal "Make conference participation fun" the three contributing intentions all contributed partially denied. The participant decided the value was unknown because "I'm not sure how any of these directly related to fun". It seems this would lead to a conclusion that the model is incomplete or inconsistent with the mental model of the participant, and thus needs to be changed, but no changes were made. In another type of example the participants treated the model and judgment situations as an oracle, deferring to the explicit model, "it's telling me that it's weakly satisfied".

A tentative conclusion is that correcting the model and producing questions relies on more extensive knowledge of the syntax, and may require explicit training in detecting differences between physical and mental models. Further studies could continue to test these hypotheses, in different situations, for example with an experienced modeler or in an industrial setting.

12.2.4.1 Additional Findings

In addition to findings supporting or denying our initial hypotheses, our qualitative analysis produced other categories of findings, resulting in new tentative hypotheses.

Model Interpretation Consistency. When examining the differences between ad-hoc and systematic analysis, we can see that some participants using ad-hoc analysis made use of the analysis labels and performed some form of label propagation (2/5), while others explained the answer to the question over the model without propagating (1/5), while some participants did both in the same study (2/5). The i* training received by all participants contained an explanation of evaluation labels.

Because the i* Framework was defined in such a way as to leave room for interpretation of its symbols and syntax, by creating systematic procedures we extend the definition of the language, making its meaning more precise. It could be argued that the interpretation used by the analysis procedures is not the best/most obvious; however, what is more important is that i* users and evaluators make consistent and similar interpretations of the model. Thus we are interested in whether or not the participants are consistent with each other (and themselves). Collected evidence shows a variety of interpretations of the model expressed via the propagation of evaluation labels, showing that ad-hoc propagation can be inconsistent among evaluators. For example, one participant interpreted the AND decomposition intentions as having to be at least weakly satisfied for the parent to be satisfied (in the procedure they would have to all be satisfied). In the same model, the participant decided that one intention in another AND decomposition was necessary for the satisfaction of the parent, but the other was optional. In several other cases propagation was consistent with the rules of our procedure. Future studies could ask participants to explicitly propagate in order to collect further examples.

Coverage of Model Analysis. Further analysis of the difference between ad-hoc and systematic analysis revealed significant differences in the coverage of analysis across the model. Subjects who used ad-hoc analysis considered the effects of far fewer intentions and actors in the models. For example, one of the participants who did propagation without systematic analysis ignored the links between the actor under analysis and another actor entirely. Several participants when propagating manually forward or backward only propagated one level or one link jump without continuing to consider the affects of other factors in the model. When participants did not propagate at all they often missed the effects of various links or intentions in their verbal analysis. For example, when considering the satisfaction of goals related to Attendee experience without propagation, a participant only looked at contributions from the Sustainability Chair and did not acknowledging positive effects from goals within the Local Chair.

Although use of the explicit analysis procedures increased the coverage of the analysis, it did not ensure complete coverage. Depending on the choices for initial values, the propagation results often did not cover the entire model. Most participants did not see any problems with such incomplete propagation. If propagation is to be complete more often, more training concerning the selection of initial values and the interpretation of analysis results is needed.

Model Completeness and Analysis. Several participants made interesting comments about the relationship between model completeness and the effectiveness of model analysis. In the Inflo case study, the participants felt that analysis was not useful until the model reached a sufficient level of completeness. One individual participant thought that the study should urge people to make a more complete model before analysis. Another participant said that the model would have been much better if there had been more time to work on it, yet this participant finished creating the model before time was up. For this participant, the analysis revealed that model was incomplete. Another participant, when applying analysis, noticed that the model had no negative links. We can conclude that analysis may be more useful for answering domain questions when the model is complete, but that analyzing over an incomplete model has the potential to reveal its incompleteness.

Inconsistent Judgments. Study results provide evidence that inexperienced modelers made inconsistent judgments. Although the participants of individual case studies followed a think-aloud protocol, we did not consistently directly ask the participants about the reasons for their decisions. However, some hints can be gained from their dialog. We can find several examples where the participant seemed to override the results provided by the procedure with their own judgment. For example assigning an unknown label "I don't see that as very important" when all incoming evidence was not unknown. Or giving the intention a fully satisfied label when the incoming evidence was one partially satisfied label and one partially denied label (implying the denied contribution was not important). The results contain other examples where the participant accepted the analysis results as an oracle, not to be questioned, e.g., "it's telling me that it's weakly satisfied". These findings motivated the creation of human judgment consistency checks described in Chapter 10.

12.2.5 Study Design Selection

Several study design choices were available, the most applicable of which being controlled experiments, action research, or case studies. An experiment would have required the isolation of as many control variables as possible in order to convince the reader that the results in terms of dependent variables (for e.g., model changes, questions asked) followed from the manipulation of independent variables (using or not using the procedure, analyzing your own or others models). In the case of goal model analysis, many variables exist which are difficult to control, including: the participants experience with i* and other goal modeling frameworks, their experience with goal model analysis, their experience and openness to modeling in general, their industry experience, and the nature and subject matter of their education. Given that we want to use participants with some i* experience, the second barrier to the application of experiments is finding enough participants to produce statistically significant results. Despite the popularity of i* in research ("Fifth International i* Workshop," 2011), in practice it is not widely used, and a large pool of i* users is not available.

Action research was a further alternative, similar to the types of case studies performed in most work which introduces goal model analysis procedures (e.g., Franch, 2006; Giorgini et al., 2004). The forward interactive procedure used in this study has already been applied to the large social service case study, producing results which led to the formation of the initial hypotheses. Although future studies of this type are useful, we believed it would be advantageous to collect evidence from multiple cases, in an effort to collect a greater quantity of qualitative data. Case studies are useful in that they can provide evidence not only to confirm the existence of hypotheses, but also to explain why such phenomena occur, particularly useful in cases with many confounding variables.

12.2.6 Threats to Validity

Several threats to the validity of our studies exist.

Construct Validity. We used several measures to test our hypotheses. To test analysis capabilities we looked at how participants were able to use the model to answer questions, whether they could apply some default questions to the models, and whether they could create and analyze their own questions over their own models. However, it was challenging to measure the difficulty participants had in performing these tasks. Often it was hard to determine if the participants were able to take analysis results and use them to draw conclusions over the domain.

To measure model iteration, we counted changes made to the model, or in some cases suggested changes. However, it is difficult to know if these changes are always beneficial. To measure elicitation, we collected questions asked over the model domain during the study. However, classification of questions versus comments can be subjective, and not all domain questions asked over the model would realistically lead to further elicitation. We used a follow-up question to measure improvements in domain knowledge. However, it is difficult to isolate whether analysis was the source of improved understanding and not simply reading or creating the models.

All other exploratory hypotheses are measured using the collecting of qualitative data. This collection can be subjective, although we battled this subjectivity to some degree by having more than one person involved in the data analysis, and by performing systematic classification of qualitative observations.

Internal Validity. We must show that the design of our study adequately tests the initial hypotheses. The extra analysis training given to participants using explicit analysis may have affected the results, although these participants didn't make any more model changes or ask any more questions. Although the study facilitator tried to encourage honest opinions, the presence of one of the authors in all study sessions may have influenced the results. The think-aloud protocol may have affected participant actions, avoiding actions they could not justify. Some of the participants were not comfortable with the think-aloud protocol, and were quiet, making it hard to understand the motivations behind their actions. It is possible that the choices of model domains influenced results, with the domains being too unfamiliar or familiar.

External Validity. As our study used upper-year undergraduate or graduate students as participants, it is possible that results may not generalize to other groups with less technical background. As our studies used the i* framework and interactive analysis procedures, it is questionable whether the results generalize to other goal modeling frameworks or analysis procedures. We believe that results are applicable to frameworks which have a syntax similar to i* (Tropos, GRL). However, it is unlikely that results will generalize to fully-automated analysis procedures.

Reliability. The study was administered by someone with expert knowledge of i* and i* analysis. If the experiment was repeated with someone with less i* or analysis knowledge, the quality of the training or of questions answered in the study may differ, and so may the results. The researcher in question is the creator of the analysis portions of the OpenOME Tool and the Analysis Methodology in question. Some of the potential bias was avoided by having each participant either use or not use the procedures, avoiding an unintentional promotion of one over the other. Every effort was made to avoid influencing the participants during the study; however, it is difficult to avoid all bias or potential effects in such cases.

12.2.7 Study Contributions

In this study we applied interactive i* analysis in ten studies with individual participants and one group study with the aim of testing existing hypotheses concerning the benefits of analysis, and discovering new knowledge about interactive goal model analysis. Results of this study address several of our requirements for early RE agent-goal model analysis. Despite the small participant sample size, the results are interesting, and not as anticipated. The results can be summarized as follows:

- **Usability and Comprehension.** Participants were generally able to apply and understand analysis, attesting to a sufficient level of procedure simplicity (R15) and the ability of the implementation to hide some of the complexities of the procedures (R16). Some difficulties were encountered in analysis comprehension, especially when it came to starting the analysis or understanding backward analysis conflicts. These issues have been addressed in Chapter 9 and Chapter 10, and are further tested with follow-up studies in the next section.
- **Analysis.** Both systematic and ad hoc analysis can be useful for answering analysis questions over the domain, although training is needed to apply initial analysis values and interpret the results (R13). We hypothesize that analysis questions were too artificial,

with students not having sufficient buy-in to engage in the analysis process, and inflo participants not driven by analysis for the sake of analysis, i.e. not having pressing analysis questions of their own.

- **Model Iteration.** Both systematic and ad hoc analysis prompted small amounts of iteration over the model, generally improving model completeness and accuracy (R4). We hypothesize, as discussed, that participants were not driven to improve the model by realistic factors.
- **Elicitation (R6).** Both systematic and ad hoc analysis prompted a small number of questions over the domain. The iteration and elicitation effects observed in previous studies may require explicit training in adjusting the model to match the analysts mental model, and using the model to reveal gaps in knowledge.
- **Domain Knowledge.** Both systematic and ad hoc analysis leads to a better understanding of the domain.
- **Model Interpretation Consistency.** Ad hoc analysis will often use interpretations of the model which are inconsistent within one analysis and amongst modelers. Use of systematic analysis promotes a consistent interpretation of the model (R7).
- **Coverage of Model Analysis.** Systematic analysis increases the coverage of intentions and actors considered in answering analysis questions. This phenomenon helps to increase the reliability of analysis (R14).
- **Model Completeness and Analysis.** A certain level of model completeness may be necessary for effective analysis. In some cases analysis may reveal the incompleteness of the models.
- **Inconsistent Judgments.** Some inconsistencies were noticed between the judgments made by users and the model. Users also had trouble remembering previous judgments. These difficulties were addressed by the judgment consistencies introduced in Chapter 10, as well as a list of past judgments for each intention as described in Chapter 11. Future studies should test these newer interventions.

12.2.8 Follow-up Studies Testing Visualization Mechanisms

In order to test the practical utility of the visualizations described in Chapter 9, we performed five follow-up studies using participants from the initial eleven studies described in the previous section. Three participants repeated analysis over small models they had created, one participant analyzed a large model created by others (Fig. 4) and the last participant analyzed the model created in the inflo case study, Fig. 2 and 6, (the participant had helped create this model). Each session lasted 30 minutes to an hour. Participants were specifically asked to comment on the new interventions: Do the leaves/roots highlighted in the model make sense? Can you understand why there is a conflict? Participants were paid $20 for their time, and results were not shared with anyone who could affect course or academic standings. Session audio and screen movement were recorded. A facilitator was present in each session directing study steps and making observation notes. Data collected was analyzed qualitatively, as in the previous studies, classifying observations into related categories or theories. Threats to the validity of the study designs are discussed in the next section.

Results of the studies are analyzed qualitatively to gauge the utility of the interventions, suggest improvements to the new visualizations and find future directions for visual support of interactive goal model analysis.

12.2.8.1 Results and Discussion

In the following section we summarize results from the validation studies, including suggested improvements to the new visualizations. Threats to the validity of the studies are summarized and alternative study designs are considered.

12.2.8.1.1 Leaf and Root Intention Highlighting

Generally, reaction to root and leaf highlighting was positive, with participants understanding the results of the automatic highlighting. A few of the highlighted roots and leaves were surprising to participants, but upon examining the links in the model more closely, they were able to determine why specific intentions were leaves or roots. The participants' surprise at the identification of some leaf/root intentions can be attributed to the difference between global and local roots and leaves. The definition in Section 5.4.1 describes global model leaves and roots, but it is also possible to identify intentions which are only leaves and roots inside of an actor, ignoring incoming or outgoing links. Often participants thought that these intentions should also be considered leaves and roots.

Because we encouraged participants to use the automatic highlighting only as a suggested starting point for analysis, in a few cases participants decided to add initial analysis values to local leaves and roots in addition to global leaves/roots. Future improvements could highlight both global and local leaves and roots automatically, perhaps using different visual cues to distinguish between global and local.

One participant stated that leaf/root highlighting was not useful, because of the small size of the model and because they had already successfully identified roots in the last round of the study. We can hypothesize that this visualization is less useful for small, familiar models.

Once leaves and roots were identified by the application, participants had an easier time selecting initial values for analysis when compared to the previous study rounds. Typically, participants went from highlighted intention to intention, deciding what value the intention should have in the current analysis question.

In the inflo case, when leaves or roots were identified, this prompted changes, adding more incoming contributions to some sparsely connected roots. In several of the studies, analysis over the model began with more initial analysis values when compared to the previous study rounds. It can be argued that adding extra constraints or initial values to the analysis produces richer, more useful results over the model.

Finally, an additional unexpected benefit emerged when in a few cases root and leaf highlighting prompted changes to model layout. Roots were moved to the top of actors while leaves were moved to the bottom, if they were not already in these positions. To be fair, most of the layout changes were prompted by the facilitator, but in all cases the participants agreed with these changes, with one participant saying it made the model more organized, making it easier to see the structure of the model.

12.2.8.1.2 Conflict Highlighting

Results concerning conflict highlighting show that this intervention is helpful in understanding model conflicts; however, a considerable knowledge of i* modeling and analysis is needed to completely understand the causes of the conflict. Often the participants were not able to understand the reasons for the conflict on their own, even with highlighting. In these cases the facilitator had to use the highlighting and assigned intention values to explain the underlying reason for the conflict to participants. This result echoes the results previous studies described in this chapter, although interactive analysis is helpful, in order to receive the full benefits, including model iteration and further elicitation, relies heavily on i* and analysis experience, or the presence of an experienced facilitator.

Despite the need for a high level of i* knowledge, highlighting of conflict intentions made it much easier for the facilitator to understand and explain conflicts in the model. All participants indicated that conflict highlighting was helpful in some way.

In several cases conflict results revealed interesting tradeoffs in the model, prompting the participant to make tradeoffs in their analysis decisions. For example, in one individual model conflict highlighting revealed a tradeoff between Networking with friends and Get a good job, with choices over Study Hard and Do an internship. In another study, conflict highlighting revealed a tradeoff between Distributing Materials for the Publicity Chair and Low cost for the Treasurer, with the means to distribute materials increasing cost. In this case the participant lowered the analysis target value for low cost from satisfied to partially satisfied and selected only one means of Distributing Materials.

In some cases understanding of conflicts prompted changes in the model, although in these cases the changes were suggested by the facilitator. For example, in one individual study, in the first run of the backward analysis, a conflict was found immediately without human judgment. The facilitator suggested that this result was due to the "all or nothing" nature of some decomposition links in the model, suggesting that these links may be better represented by contribution links. In this case the participant agreed and changes were made. The next run of the analysis found a solution in the model.

Results also revealed that the logical source of the conflict (defined in Section 9.2.3) is not necessarily equivalent to the "conceptual source". Here the logical source is the most immediate or direct cause of the conflict (e.g., PS and not PS) as reported in the UNSAT core, while the conceptual source is the construct in the model which is the originating source of the conflict, i.e. if this were removed/changed, the conflict would disappear.

We see an example of this in the inflo conflict example in Figure 103, repeated with zoom from Figure 57. Here, the conflict sources include several tasks (showed in the zoomed out view), while the conceptual source is the hurt link from Type checking and conversion to Flexibility (purple circle).

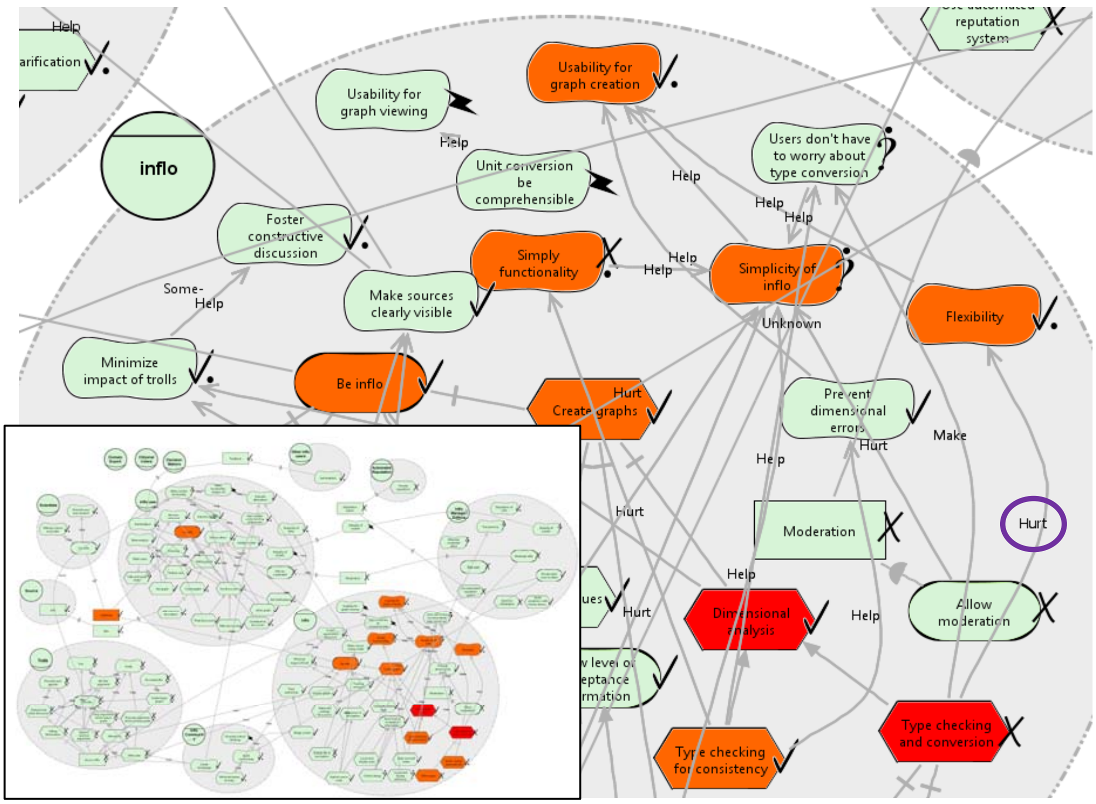


Figure 103 View of Conflict Highlighting in the i* Model Resulting from the inflo Case Study

In another example in one of the ICSE Greening models (Figure 104 repeated with zoom from Figure 73), the logical source of a conflict was Update web page while the conceptual source was the unknown link from Prepare materials to Attractive materials which constrained Prepare materials from having a denied value (purple circle). In these cases it may not make sense to remove or change the conflict source construct in the model – perhaps Type checking and conversion really does hurt Flexibility. In such cases a conflict in the domain is revealed. Future work should investigate methods to suggest conceptual sources for model conflicts, perhaps highlighting negative or unknown links along the path of the conflict.

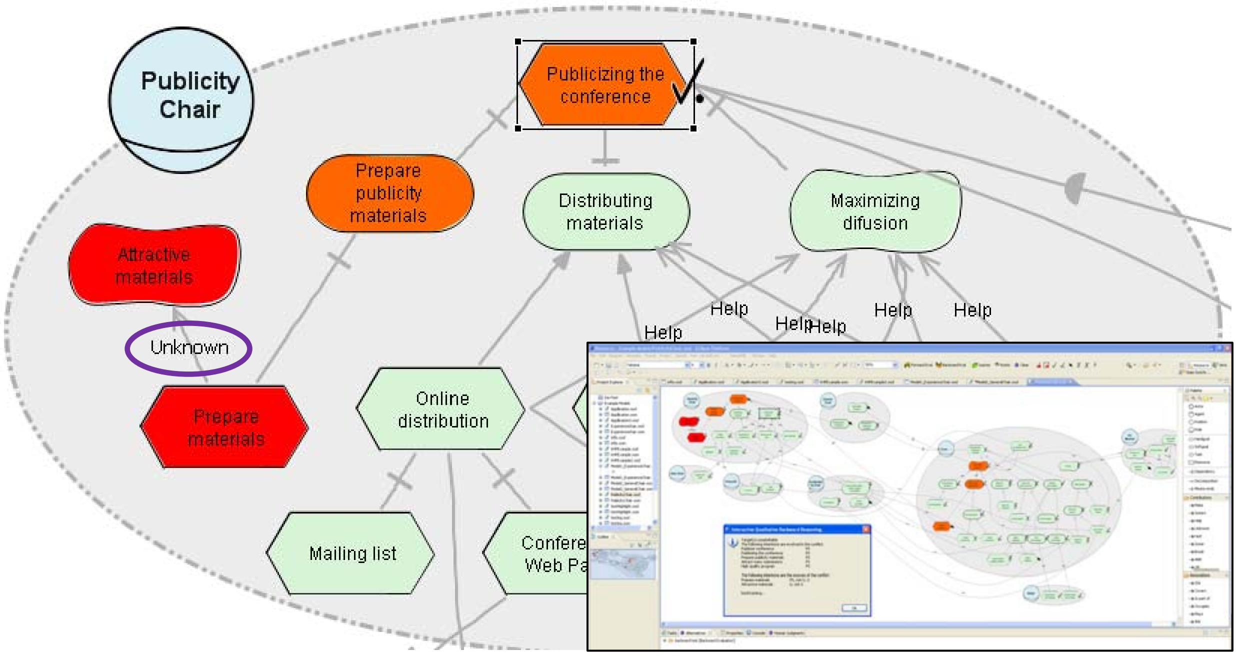


Figure 104 Example Backward Analysis Conflict in OpenOME

Results also revealed that conflict highlights should be left in the model for a longer duration of time, perhaps until after the user has completed the next backtracking stage. One participant mentioned that they were having difficulty remembering where the previous conflicts occurred when they were making judgments. They suggested leaving "traces" or some type of list of past conflicts.

12.2.8.2 Threats to Validity

Although the results of our studies were a useful first step towards testing the utility of the new visualization to support interactive goal model analysis, several threats to validity exist. Threats to the validity of this study are similar to the threats described for the original ten individual and one group case study. As the subjects of our study were all students, all with some exposure to modeling in systems development, it is possible that results would not generalize to participants with different backgrounds or experiences. It is possible that the subject matter of the models or the size of the models may have affected results; however, we have tested the visualizations over models in three domains with a variety of sizes. Finally, the study designers and facilitators are the inventors of the analysis procedure and visualizations, possibly introducing bias via their presence in the room, their guidance as facilitators, and through their interpretation of the results. We have tried to minimize bias whenever possible, encouraging participants to be honest, and reporting both positive and negative results.

Future work could take an action research approach to testing visualizations. Interactive analysis could be applied to a more realistic problem in an organization, noting any benefits of the visualizations and recording ideas for improvements.

12.2.8.3 Study Contributions

Results of follow-up studies testing visualizations showed that the presence of a facilitator may be necessary to understand highlighted conflicts, or to identify areas of useful model changes identified through the visualizations (R2). However, the visualizations helped users to more easily identify starting points for analysis, asking more complete analysis questions, helping to improve analysis power and the reliability of analysis (R13, R14). Explicitly identifying leaves and roots can help to improve the layout of the model, helping to handle model complexity. Conflicts were more easily explained to participants with the presence of highlighted intentions (R2), and interesting tradeoffs in the domain were identified by participants.

12.3 Related Work: Case Studies Using Intentional Modeling

We can find examples of studies applying intentional modeling in repeated case studies or experiments. In (Stirna, Persson, & Sandkuhl, 2007), the authors describe multiple participatory cases to illustrate guidelines for participatory Enterprise Modeling (EM). Related work uses two studies to derive conclusions and recommendations about participatory modeling and tool support (Persson & Stirna, 2002). An interesting conclusion of this work is that organizational modeling requires an expert in such modeling. Our findings concerning the need for more extensive i* and analysis training reflect this finding; however, we believe it is too restrictive to say that i* and associated analysis should only be used with an expert present. Existing work shows that even i* novices who misuse the notation benefit from its use (Elahi, E. Yu, & Annosi, 2008), and our individual participants generally increased their knowledge of the domain through modeling and analysis.

Work similar to the pattern study described in this chapter, evaluated patterns developed in the EKD (Enterprise Knowledge Development) method via workshop experiments involving experienced professionals (Rolland et al., 2000).

12.4 Framework Validation Conclusions

In this chapter we have described several studies which test both forward and backward analysis. Initial examples, a large counseling service study, and an exploratory experiment tested the manual application of forward analysis. The implementation of forward and backward analysis was tested via the application of individual case studies and the inflo group case study. Follow-up studies tested the effectiveness of visualization techniques. We summarize the contributions of these validation studies using our agent-goal model early RE requirements. Additional findings beyond these initial requirements have also been found, and are summarized here.

R1 Scalability and R2 Analysis Comprehension. Manual application of the forward procedure to multiple case studies has shown the procedure to be scalable over models of a reasonable cognitive size, with model results understandable in terms of the domain. Examples of larger models which were able to be analyzed and understood include the TC models, the third stage counseling models, the inflo model, and the ICSE Greening Examples. However we have also shown the limits to procedure scalability and analysis comprehension through the creation and analysis of very large models in the counseling service studies.

We also discovered that explicitly identifying leaves and roots can help to improve the layout of the model, helping to handle model complexity.

Individual participants were generally able to apply analysis. However, when using the larger ICSE Greening Models, their lack of familiarity in the domain and the size of the models hindered their understanding. Using smaller models created by the participants about student life improved their ability to understand analysis.

We have attempted to address difficulties in starting analysis or understanding conflicts with interventions described in Chapter 9 and Chapter 10, with follow-up studies of visualization mechanisms showing that these mechanisms were helpful in analysis, although comprehension issues were still present. Future studies should take these visualizations and other comprehension mechanisms further.

P3 Partial Automation. Evidence collected in the studies supports the presence of our requirement for partial automation (P3), as the procedure was difficult to apply manually on large models.

R4 Prompt Model Iteration. While conducting examples such as the TC study, we noticed the ability of the procedure to prompt model iteration, improving the quality and completeness of the model. The effect was noticed again as part of the large counselling service study, although we did not begin to measure changes made until the third stage of the study, where we discovered that analysis conducted by two different individuals caused fairly significant changes to the pattern and contextual models in the pattern study.

As a result of these observations, we conducted an exploratory experiment to test model iteration caused by analysis. Results of the experiment indicate that the evaluation procedure does prompt model iteration, but it is unclear whether this iteration occurs because of the systematic nature of the procedure, or just a careful examination of the model brought on by analysis.

More in-depth studies with a larger number of individual participants also produced mixed results in terms of model iteration. Both systematic and ad hoc analysis prompted small amounts of iteration over the model, generally improving model completeness and accuracy. We hypothesize, as discussed, that participants were not driven to improve the model by realistic factors.

Results of follow-up studies testing visualizations showed that the presence of a facilitator may be necessary to understand and identify areas of useful model changes identified through the visualizations. These findings helped lead to the judgment inconsistency checking in Chapter 10 as a means to embed i* expertise into the analysis implementation. The effectiveness of human judgment inconsistency checking in prompting model iteration should be addressed in future studies.

R7 Provide Model Interpretation and R14 Reliable Analysis. Results of our studies showed that ad hoc analysis will often use interpretations of the model which are inconsistent within one analysis and amongst modelers. Use of systematic analysis promotes a more consistent interpretation of the model. This would, in turn, help to promote more reliable analysis.

R9 Accommodate High-Level Domain Information. All of the studies demonstrated the analysis procedures ability to accommodate high-level domain information such as security, confidentiality, quality of counselling, and a green conference, drawing conclusions over concepts which are hard to define formally.

R12 Iterative Methodology. We were able to test the iterative methodology described in Chapter 8 in the second round of our individual case studies. Although we did not see any significant change in model iteration using this methodology, we wrote the methodology trying to be neutral in terms of encouraging model changes. In other words, we did not want to bias the results of the study by telling participants to iterate over their model, but only suggested to do so if they felt the need. Future changes to the method could place more clear emphasis on iteration and model improvement.

R13 Analysis Questions. Examples of manual forward application show that forward analysis can answer useful domain questions. Example analysis questions answered by the procedure include Will TC Work? and Which technologies can help provide online counselling, and why?

In our exploratory experiment, we note that participants made changes in their analysis results when applying the evaluation procedure after applying ad-hoc analysis. This attests to the analysis power of the procedure, producing different results than the users would obtain without systematic analysis.

The individual case studies show that both systematic and ad hoc analysis can be useful for answering analysis questions over the domain, although training or further implementation mechanisms are needed to help apply initial analysis values and interpret the results. Implemented visualizations address these issues to some degree, helping users to more easily identify starting points for analysis. Future studies should be devised to test analysis in situations where questions are less artificial, and motivated by realistic factors.

R15 Simple Analysis Procedures and R16 Tool Support Hides Complexity. Application of the forward procedure manually to many examples attests to a sufficient level of procedure simplicity, although the examples showed that tool support providing some level of automation was needed. Although the author was the primary evaluator in most of the examples, in the counsellor service and exploratory experiments, other individuals familiar with the respective domains were able to apply forward analysis manually.

Participants in the individual case studies were able to apply and understand analysis using the OpenOME tool, although some usability issues were discovered, addressed partially by visualizations, judgment checks, and other implementation mechanisms.

R6 Prompts Further Elicitation. Initial examples and case studies did not explicitly capture (i.e. make counts) of the iteration prompted by analysis. In the initial examples, analysis sometimes raised questions which prompted the evaluator to go back to the model sources for more information. In the counseling service study, stakeholder time was scarce, so often questions which arose from analysis (e.g., what do they mean by confidentiality?) were discussed amongst the analysts.

The exploratory experiment showed that careful examination of a model through ad-hoc or through systematic analysis leads to further elicitation. In the individual case studies, both systematic and ad hoc analysis prompted a small number of questions over the domain. We hypothesize that the more training would help users to recognize gaps between their mental model and the physical i* model.

Increase Domain Knowledge. All of the examples applying manual forward analysis increased the evaluators understanding of the model and the domain. However, such an increase is difficult to measure, especially as most of the studies were performed before this particular hypothesis was articulated.

The follow-up questionnaires for the exploratory experiment revealed that all participants claiming they gained a better understanding of the domain using the evaluation procedure than with no procedure. The majority (7/10) individual participants claimed that analysis improved their domain; however this effect occurred with both ad-hoc and systematic analysis. Several participants in the inflo study pointed out the ability of analysis to point out interesting tradeoffs in the domain.

Coverage of Model Analysis. In addition to existing hypotheses, studies revealed that systematic analysis increases the coverage of intentions and actors considered in answering analysis questions. This includes asking more complete initial questions over the domain, facilitated by root and leaf visualizations. We claim that this effect will help to increase the reliability of analysis results.

Model Completeness and Analysis. Results of the validation studies also showed in several cases that models were incomplete. In manual forward application to strategy documents, analysis revealed that i* models created from documents were not well connected. Several participants in the individual studies pointed out missing information as part of the analysis process, and participants the inflo study noted that a certain level of model completeness may be necessary for effective analysis.

We summarize the contributions of this chapter over the model in Figure 105. In the next chapter, we summarize cumulative contributions of this and the previous chapters.

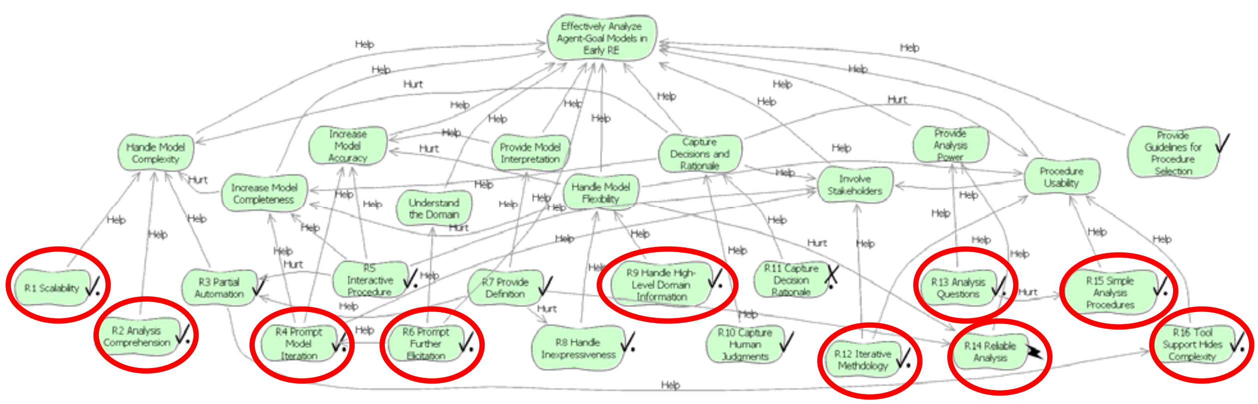


Figure 105 Satisfaction Analysis for the Requirements for analysis of Agent-Goal Models in Early RE based on a Combination of Existing Work

Chapter 13 Contributions, Limitations, and Future Work

In this chapter we review the challenges of early RE analysis, the limitations of existing work, and the requirements for early RE model analysis. We assess to what degree the framework described in this work has met these requirements and challenges, including a discussion of what issues are left to be addressed in future work. Limitations to the Framework are described and future work in the area of interactive goal model analysis for early RE is outlined.

13.1 Summary and Contributions

13.1.1 Motivations Summary

In Chapter 1 we have identified the need for iteration over models, finding effective methods to prompt improvements in model quality and more effective abstractions. These methods are especially needed in the area of early requirements, where the domain is complex, socio-technical, and involves the introduction of new system or system features. It is more challenging to accurately model what will be than to model what is.

Intentional goal models have been selected as an effective way to capture important aspects of the domain in early requirements, specifically goals, actors, and their inter-relationships. Systematic analysis over such models can allow stakeholders and analysts to improve the quality of the model, improve domain knowledge, and answer useful questions over the domain. Although analysis over goal models can provide several benefits in early RE, such models can be challenging to analyze for several reasons. Early RE models capturing socio-technical domains can quickly become complex, containing many actors, goals, and relationships, making analysis challenging. The social and “to-be” nature of the models means that it is difficult to know when models are sufficiently complete, or how this completeness will affect analysis. Similarly, it is difficult to argue that an early RE goal model is ever accurate; instead we can aim for sufficient accuracy, searching for methods which help to improve accuracy to an adequate level.

Understanding a new domain in the early stages of requirements analysis can be challenging. Using agent-goal models for this purpose can help, but it is difficult to know whether domain knowledge is sufficient to select and specify an effective technical intervention. In order to capture important social and non-functional aspects of the domain, agent-goal models and their respective analysis procedures must be sufficiently flexible. They must support "fuzzy" concepts such as softgoals and contribution links. This flexibility can make it challenging to interpret models, or to ensure that interpretations remain the same amongst multiple parties. Although previous applications have argued that goal models can be useful for early RE decision making (e.g., Amyot et al., 2010; Giorgini et al., 2005), it is often difficult to trace the rationale for decisions made over tradeoffs contained within agent-goal models.

Although agent-goal models endeavor to incorporate real-world concepts (goals, agents, etc.) their use still involves a learning curve. As such, it is challenging to gain sufficient stakeholder involvement in the modeling and analysis process. It is challenging to support a wide range of analysis questions over agent-goal models in early RE, especially without sacrificing procedure usability or modeling language flexibility. As early RE aims to involve stakeholders in a time-sensitive process of elicitation and discovery, sufficient procedure simplicity is a key to successful application. Finally, the large number of available procedures for early RE goal model analysis makes procedure selection challenging. We repeat our summary of the challenges for agent-goal model analysis for early RE in Figure 106.

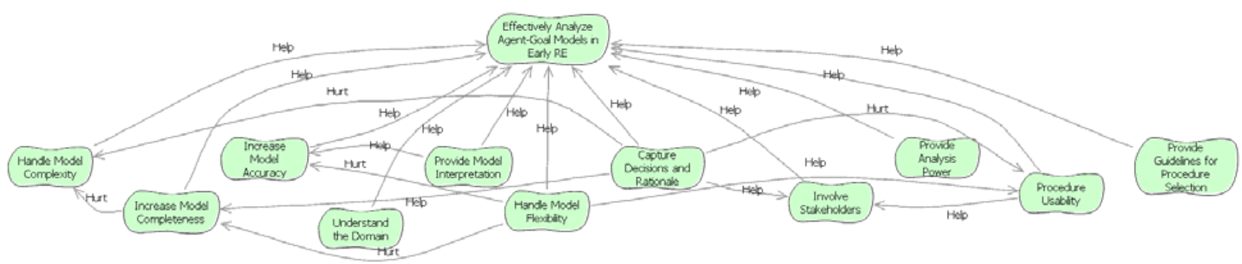


Figure 106 Summary of Challenges for Agent-Goal Model Analysis

In order to address the challenges of goal model analysis in early RE, we have operationalized the challenges/analysis goals into more detailed goal/requirements, as detailed in Chapter 3 and summarized again in Figure 107.

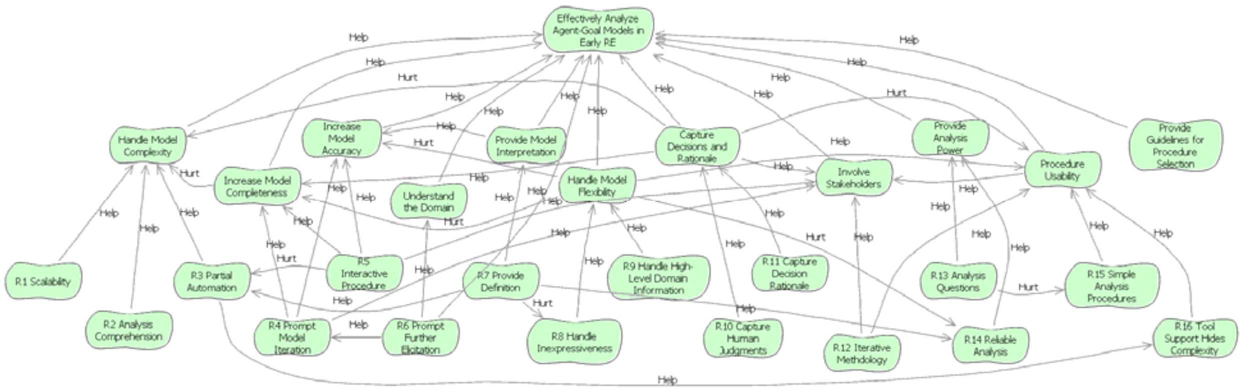


Figure 107 Requirements for analysis of Agent-Goal Models in Early RE

Many techniques for the analysis of goal models exist, using a variety of technical approaches including qualitative and quantitative satisfaction analysis, metrics, planning, simulation and model checking. We have provided a thorough summary of such work in Chapter 2.

Existing work focuses on the analytical power of goal models, often assuming model completeness and accuracy. Such work often assumes the presence of metrics or formal definitions of goal-oriented concepts, and often requires additional information such as temporal ordering or prioritization, information which is difficult and laborious to accurately elicit in early RE. In Chapter 3, we have compared existing analysis approaches against our elicited requirements for early RE analysis. We conclude many existing procedures may not be sufficiently scalable and that existing procedures do not aid in analysis comprehension. It is unclear if existing procedures are able to sufficiently prompt model iteration or elicitation and is unknown if such procedures are sufficiently simple, or produce reliable analysis. We repeat our summary of existing procedures in light of our early RE agent-goal model analysis requirements in Figure 108.

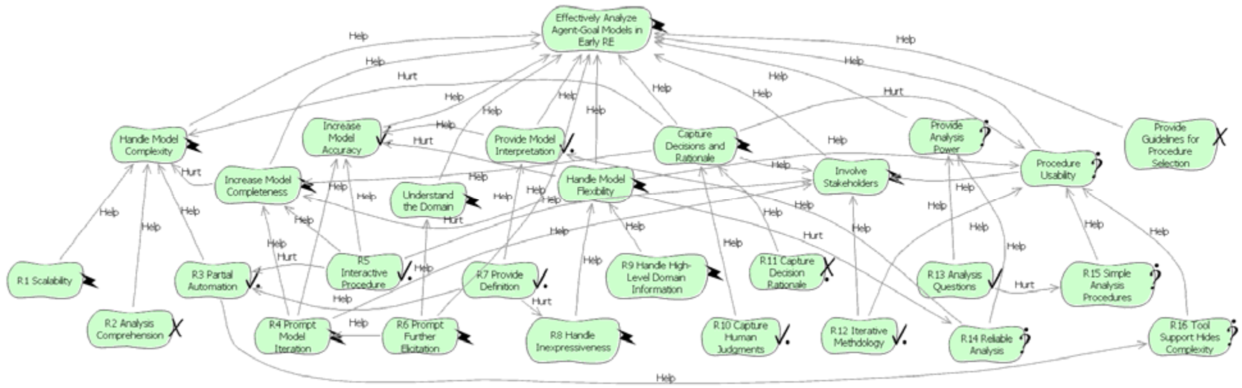


Figure 108: Satisfaction Analysis for the Requirements for analysis of Agent-Goal Models in Early RE based on a Combination of Existing Work including Conflicts and Synergies

After assessing existing goal model techniques and analysis procedures in light of our framework goals, it was decided to base our framework on the interactive analysis procedure over i* models described in Horkoff (2006), as it made progress towards our goals for early RE agent-goal model analysis. This procedure made claims concerning iteration, interaction and elicitation, including the use of human judgments (summary repeated in Figure 109). The remaining sections of this work were aimed at producing analysis, introducing mechanisms, and administering studies which addressed the shortcomings of this procedure when compared to our framework goals.

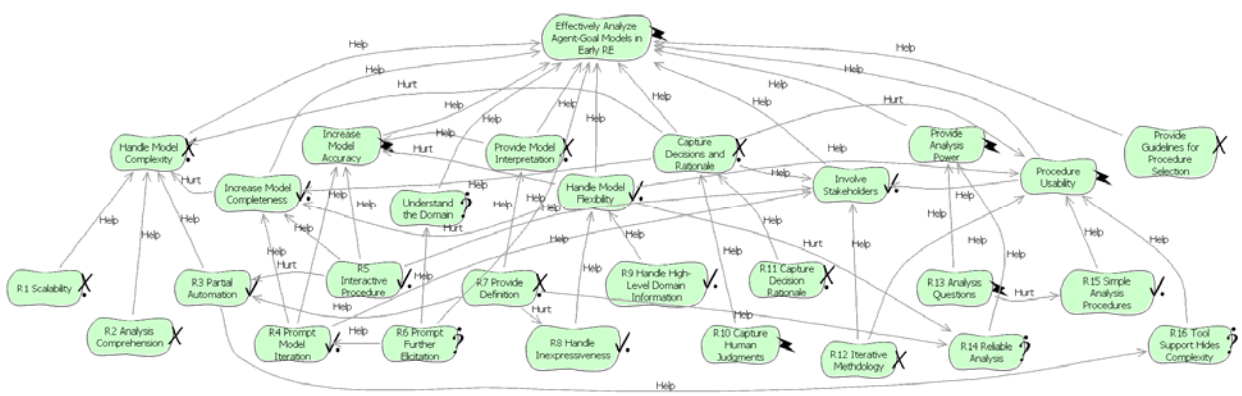


Figure 109: Satisfaction Analysis for the Requirements for analysis of Agent-Goal Models in Early RE based on the contributions of Horkoff (2006)

13.1.2 Framework Component Contributions

In the following, we review our requirements for early analysis of agent-goal models in RE, and evaluate the contributions of our framework against these requirements.

Model Complexity. From this challenge, we have derived requirements for scalability, comprehension, and tool support, as follows:

R1 Scalability: *The analysis framework must contain techniques which are applicable over large models.*

We have addressed scalability in our exploration of the running time of the forward procedure in Chapter 6, the backward procedure in Chapter 7, tool support design and running time tests in Chapter 11, and validation studies in Chapter 12. The running time of the forward procedure was shown to be a factor of *n*, the size of the model, given our restriction over the number of value fluctuations. In Chapter 7 we have shown that the running time of the procedure is bounded by the running time of the SAT solver, n^2, where n is the size of the model, and 6^q, where q is typically less than 10, $O(6^q(ln^2 + n(\text{zChaff})))$. The procedure terminates given that the user does not continually repeat judgments. The implementation of the procedure has shown to be scalable to models of a reasonable size in Section 11.4.2. Models larger than this would be no longer cognitively scalable for manual creation and analysis comprehension. Finally, extensive application of the procedure in case studies and experiments has pushed the boundaries of procedure scalability, especially over manual application of the forward procedure. Studies have shown that the procedure can be applied manually to models of a reasonable size. Although newer users had some difficulty analyzing large models that they did not create, study results show that users are generally able to use the OpenOME implementation to analyze medium to large sized models, with minimal training. Overall, although the procedure has shown to be reasonably scalable over several examples, optimizations could still be made, and we judge this requirement to be partially satisfied.

R2 Analysis Comprehension: *The analysis framework should contain methods to support comprehension of analysis results over complex models.*

We have addressed this requirement primarily through the visualization mechanisms in Chapter 9, and testing applications in Chapter 12. Large example applications of the forward procedure showed that analysis results were challenging to understand over large models. We have addressed this challenge to a certain extent by making recommendations concerning scoping as part of our iterative methodology, aiming to restrict the size of the model. Further studies showed that users had difficulties knowing where to start analysis, where to find the focus of current judgments, and how to interpret analysis conflicts. Visualizations mechanisms were introduced to specifically address these challenges. The first and last mechanism were tested in a series of follow-up studies, showing that although the visualizations made analysis easier to apply, new users still needed guidance from a model expert. As we have made some progress towards this goal, we judge it to be partially satisfied, although further means to aid analysis comprehension are needed.

R3 Partial Automation: *Procedures in the analysis framework should be supported by tools which provide some automation for analysis over large models.*

As described in Chapter 11, we have provided tool support which partially automates analysis over models. Flexibility is provided to users by allowing them to make human judgments over the model as part of analysis. Otherwise, propagation through the model is handled completely by the tool. This alleviates the complexity of propagating labels through large or complex models. We judge this goal to be satisfied in our summary model (Figure 110).

Model Completeness & Accuracy. We decompose this challenge into requirements for model iteration and interaction.

R4 Model Iteration: *The framework should contain analysis methods which encourage model iteration.*

The analysis procedure in Horkoff (2006) used experiential evidence from case studies and examples to argue that the procedure prompted beneficial model iteration. We have tested these claims in this work by applying the forward and backward procedure to several more studies of various sizes, using a number of different participants. Results show that when experienced models use the analysis procedure in realistic case studies, beneficial model iteration is made. However, when using the procedure in more artificial environments, without the presence of driving domain questions, far fewer changes to the model are made. This can be due both to the artificiality of the study, and to the inexperience of the modelers. Furthermore, our experimental studies show that even ad-hoc analysis will provoke model iteration, although systematic analysis invoked further changes on the same models. Results for this goal are somewhat mixed, although we claim that in the appropriate situation - knowledgeable modelers motivated by driving question in a real domain – analysis can produce beneficial iteration. We mark this goal as partially satisfied in our summary model. Further, more realistic, studies are needed to test these claims.

We have guided further model iteration by introducing a methodology for model creation and analysis which encourages changes prompted by analysis, further addressed in the iterative methodology requirements.

Results of the validation studies lead us to create the consistency checks described in Chapter 10, aiming to help new users understand when the structure of model and their judgments are inconsistent, embedding model expertise in the tool support, aiming for increased model iteration. Future studies are needed to test if these checks are helpful in promoting model iteration.

R5 Interactive Procedures: *The framework should contain analysis methods which encourage interactive analysis.*

Both the forward procedure in Chapter 6 and the backward procedure in Chapter 7 are interactive in that they allow the user to resolve partial or conflicting information via human judgments. In the forward direction, the users are asked to decide on a resulting analysis label given multiple sources of incoming labels. In the backward direction, users are asked, given a resulting label, if there is at least one combination of incoming labels that could produce this label. In this way, procedures are interactive, encouraging the involvement of stakeholders in the analysis process. It is difficult to know whether the level of interactivity provided by the procedures is the “right” or optimal level of interactivity, therefore we mark this goal as partially satisfied, leaving room for studies which search for an optimal level of interactivity in the early RE process.

Domain Knowledge: The analysis procedures in the framework prompt an increase in domain knowledge.

R6 Prompt Further Elicitation: *The framework should contain analysis methods which reveals gaps in domain knowledge and prompts users to fill those gaps through further elicitation.*

Results for this goal are similar to those for prompting model iteration. Studies conducted as part of Horkoff (2006) and as part of the forward manual application examples in this thesis showed evidence that interactive analysis helped the user to reveal gaps in the model and gaps in domain knowledge, encouraging modelers to return to domain sources for more information. The exploratory experiment revealed that systematic analysis prompted domain questions in addition to those gathered by ad-hoc analysis. Studies applying interactive forward and backward analysis with individual participants showed some evidence of further elicitation; however, in the first rounds users were not familiar with the modeling domain, making it difficult for them to begin to fill the gaps in their knowledge. In the second, it could be argued that users were too familiar with the domain, having created the models of student life themselves. Similarly, in the inflo study, the stakeholders were the modelers, so all gaps in knowledge were immediately filled by individual participants or through a group discussion. We claim that this benefit is again conditional: if the participants are driven to learn more about a domain in which they are familiar, but not experts, as in the process of early RE exploration, we believe interactive analysis will prompt further elicitation. As we have found some evidence to support this goal, we mark it as partially satisfied.

Model Interpretation: The flexibility and inexpressiveness of agent-goal models lead to the need for a consistent and precise definition of relevant concepts.

R7 Definition: *The framework should contain a more formal or precise definition of the underlying agent-goal model framework. The analysis procedures should be formally defined in an effort to avoid divergent application or interpretation.*

Chapter 5 began to address the requirement for providing a definition of the modeling concepts underlying the analysis framework. The forward and backward analysis procedures defined in Chapter 6 and Chapter 7 were also defined formally, including definitions of useful analysis concepts such as analysis alternatives. Visualization mechanisms and human judgment checks were also accompanied by formal definitions. We judge this requirement to be satisfied in our summary model.

The results of our individual studies show that users with little i* experience using ad-hoc analysis make divergent interpretations of the model. In comparison, participants using systematic analysis implementing the precise model interpretations introduced in this work produced analysis results and model interpretations which were more consistent.

Model Flexibility: The framework should allow analysis over high-level, potentially ambiguous concepts which are not represented formally or quantitatively. We address this challenge by requiring that analysis procedures handle inexpressiveness and high-level domain information.

R8 Accommodate Inexpressiveness: *The framework should contain analysis procedures which do not require formal or quantitative definitions of key model concepts.*

The formal definition of i* provided in Chapter 5 included a reflection on i* syntax, reviewing common deviations in order to effectively balance the need to providing a precise model interpretation with the need for inexpressiveness in imprecise early RE concepts (privacy, user satisfaction, etc.). Although the concepts and interrelationships underlying the framework have been given a more precise definition, they are still flexible enough to represent ideas which are difficult to formalize or quantify. The analysis procedures and extensions defined in Chapter 7, 10, 11, and 12 are designed to work over inexpressive models, applying qualitative analysis based on initial values and model structure. We have tested the ability of the framework to apply analysis over high-level concepts arising in early RE through application to several examples and case studies in Chapter 12.

R9 Accommodate High-Level Domain Information: *The framework should contain analysis procedures which do not require detailed domain information, difficult to acquire in early RE stages.*

In defining the forward and backward analysis procedures, including visualizations and human judgment checks, we have deliberately avoided requiring additional information beyond what is typically required by i* models. For example, although many other goal model analysis procedures would require an addition of priorities, costs, time or other measures, the procedures in this work deliberately require only goal model concepts (goals, softgoals, tasks, resources, and actors) and relationships. In this way, we support a lightweight analysis over early, exploratory models. It is difficult to know if we have accommodated high-level domain information in an optimal way, in other words, are these the right concepts to support with our analysis? With this in mind, we mark this requirement as partially satisfied.

Decision Rationale. Analysis procedures should help to capture the rationale for decisions; in our case we interpret this as capturing human judgments and decision rationale.

R10 Human Judgments: *The framework should support ways to capture, store and analyze the analysis decisions made over contentious areas in the model.*

Both the forward and backward analysis procedures have been designed to allow users to enter judgments over areas of the model with partial or conflicting evidence. The implementation description in Chapter 11 describes how this judgment is stored in the tool metamodel, with several views allowing users to view judgments as part of analysis results, or over all alternatives. We have provided analysis over the judgment themselves through the consistency checks described in Chapter 10, with judgment checks currently being implemented in the OpenOME tool. We judge this requirement to be satisfied in our summary model.

R11 Decision Rationale: *The framework should support ways to capture the rationale for decisions amongst alternatives, including varying analysis results which lead to these decisions.*

We currently leave the design, implementation and testing of this requirement to future work, see the next section for more details. The framework as it is designed provides some rationale for decisions by storing analysis results; however, more work in this area is needed. We judge this requirement to be partially denied.

Stakeholder Involvement. Early RE analysis should encourage the involvement of key stakeholders for elicitation and validation of early requirements. The framework requirements for model iteration (R4) and an interactive procedure (R5) already help to address these challenges by encouraging stakeholders to interact with the analysis process, iterating over the model. We further address this challenge by adding requirements concerning a framework methodology:

R12 Iterative Methodology: *The framework should contain clear methodologies to guide the process of interactive analysis including iteration over model content.*

We have introduced a methodology for model creation and analysis which encourages iteration over the model based on analysis results. Preliminary results in Chapter 12 (the second round of the individual studies) did not show that this procedure prompted elicitation when compared to analysis using an earlier, less descriptive, methodology which did not prescribe the use of analysis as a model sanity check. Further work is needed to test the utility of the methodology, including changes which may increase its ability to encourage iteration.

Analysis Power. Analysis procedures for early RE analysis should support a variety of types of analysis, allowing users to ask several types of questions over the model, and ideally should produce accurate, sensible and reliable results.

R13 Analysis Questions: *The framework should support a variety of analysis questions over agent-goal models, including "What if?" analysis.*

By adopting the re-describing procedure in Horkoff (2006), we have allowed for "what if?"-type questions, including "what are the effects of a particular analysis alternative?", "are goals sufficiently satisfied?", and "whose goals are satisfied?". In addition, we have introduced the backward analysis procedure in Chapter 7, allowing users to ask "is it possible to achieve certain goal(s)?", "if so how?", "who must do what?", and "if is not possible, why not?". In Chapter 9 we have helped users to ask "where do I potentially start analysis?", "over what intentions am I making a judgment?", and again "if there is no solution, why not?". The human judgment checks in Chapter 10 help to answer questions such as "have I(we) changed our minds?" and "is our judgment consistent with the model (is the model consistent with our judgment?)".

We have tested the ability of the procedures to answer most of these questions in our Chapter 12 validation studies. Results found that for forward analysis in realistic studies such as the counseling service study, analysis was very helpful in comparing and assessing technical alternatives and knowledge transfer mechanisms, including allowing for "as-is" to "to-be" comparisons. The inflo study revealed that backward analysis was useful in answering some basic analysis questions which tested the sanity of the model.

Work introducing other analysis procedures (Section 2.3) has shown that a wider range of analysis questions could be asked over agent-goal models, such as "is there a plan?" or "What is the cheapest solution?". However, without expanding the procedure to consider extra information beyond what is captured in the basic version of i*, or without making the framework overly complex, we claim that we have provided for a reasonable number of analysis questions over early RE models, marking this goal as partially satisfied.

R14 Reliable Analysis: *The framework should produce results which are accurate, sensible and reliable.*

Chapter 4 addressed the reliability of existing procedures for forward satisfaction analysis, including the Horkoff (2006) procedure used as a basis for this work. Seven existing procedures were applied to several example models from the literature, with results compared and analyzed. It was discovered that the structure of early RE models (e.g., many softgoals or dependency links) had a significant effect on the results when compared between procedures. The reliability of early RE forward satisfaction analysis results was called into question, recommending analysis over early RE models be used more as a heuristic than an oracle. Results placed a greater emphasis on the use of such procedures as means to improve model quality and domain knowledge, as opposed to purely a tool for decision making.

We returned to the issue of analysis reliability in our validation studies described in Chapter 12. Here, it was argued that systematic analysis, including leaf and root visualizations, made the starting points of analysis more complete, forcing users to take into account additional model constructs when compared to ad-hoc analysis. We also discover that systematic analysis increases the consistency of model interpretations, e.g., propagation through contribution links. These factors would make analysis results more consistent or reliable when comparing results over the same model, potentially with different evaluators. Overall, we have found conflicting evidence towards this goal, and we judge it to have a conflict value in our summary model.

Procedure Usability. Although several existing agent-goal model analysis procedures may be applied in an early RE context, it is not clear if these procedures are practically usable. To ensure usability, especially if the aim is to involve stakeholders in the modeling and analysis process, procedures should be as simple as possible to apply, with as much complexity as possible hidden by tool support, and should be guided by clear methodologies (R11).

R15 Simple Analysis Procedures: *The framework should contain analysis procedures which are simple enough to be applied with minimal training.*

We have aimed to design analysis procedures which are simple from the perspective of the user (Chapter 6 and Chapter 7). Although these procedures have been described formally, it is not necessary for users to see or understand the formal definition, depending on their level of comfort with such a definition style. In essence, both the forward and backward procedures involve 1) coming up with an analysis question 2) applying labels to reflect this question 3) evaluating the model, answering judgment questions as they arise, and 4) understanding procedure results. From the user perspective, we argue the procedure steps are relatively simple.

We have tested this requirement more thoroughly in multiple studies Chapter 12. The procedure author applied the forward procedure to many examples, demonstrating its simplicity to a certain extent, although at the same time pushing the limits of cognitive model and analysis scalability. Other researchers, most of which had experience in other types of modeling, were able to apply the procedure after a session of training over example models, or after reading a training document provided (see Horkoff, (2009) for an online version of the training document). These individuals grasped the forward analysis procedure with relative ease.

In the individual case studies, participants in the first stages were given brief handouts describing i* and the procedure (for participants in the systematic procedure group). Observations showed that the amount of training was not sufficient. The next round (8/10) of participants were given a 10-15 minute training session by the facilitator (thesis author), and the final four participants were given the methodology for model analysis described in Chapter 8. In any of the participant training scenarios, participants were able to apply both the forward and backward analysis, the deficiencies were noted more in their ability to understand the meaning behind i* syntax than their ability to apply analysis. Although usability issues were noted, several of these issues were addressed in the interventions described in Chapter 9, 12, 13, and most of the issues with analysis were observed to be more at the detailed level (e.g., applying initial labels, understanding results) than the overall process or purpose of analysis. Using the evidence collected, we judge this goal to be partially satisfied in our summary model.

R16 Tool Support Hides Complexity: *The implementation of analysis procedures in the framework should encode and hide as much complexity as possible from the user*

Although we have described the i* framework, the forward and backward analysis procedure, visualizations and human judgment checks formally, the OpenOME implementation hides these details (predicates, axioms, etc.) from the user. The user sees only a visual display of the model, analysis labels, and summaries of analysis results in the alternative and human judgment views. The second half of our validation studies in Chapter 12 tested the usability and simplicity of the framework implementation in the tool, as described in our consideration of the R15 Usability requirement. Although some usability issues were noted, several of these issues have been fixed in successive iterations of the tool. Overall, the implementation was usable with minimal explanation from the study facilitator. We judge the goal to be partially satisfied. Future studies should test some of the newest implementation features, such as the human judgment window and highlighting of intentions involved in a human judgment.

Procedure Selection. The presence of many analysis procedures for agent-goal models makes it difficult for potential users to select an existing analysis procedure. Because of the wide variety of procedures available for goal model analysis, it can be difficult and intimidating, especially from the industrial point of view, to understand which method to apply in what situations. To this end, we have provided guidelines for procedure selection depending on the purpose of analysis and information available in the domain, providing guiding questions and illustrating procedure selection with examples (Section 2.7). This work has addressed our goal of Provide Guidelines for Procedure Selection in the Figure 110 model.

Finally, given the comparative analysis, mechanisms, implementation, and validation described in this thesis, we summarize the satisfaction of early RE agent-goal model requirements, in Figure 110, below. We can understand the contributions of this thesis as summarized by this model more effectively by comparing the results, first, to our cumulative summary of existing analysis procedures (Figure 7), and next, to the base work for this thesis, the forward procedure described in Horkoff (2006) (Figure 8). We provide a model integrating these results in Figure 11, with values from existing work in red, Horkoff (2006) in blue, and the contributions of this thesis in green. The reader can note that through several improvements over our requirements, including scalability, analysis comprehension, model iteration, elicitation, methodology, analysis questions, simplicity and tool support, the overall value for Effectively Analyze Agent-Goal Models in Early RE is now judged to be partially satisfied, an improvement from the previous conflict value. In the next sections, we describe the limitations of this work (i.e., why the top level goal is partially and not fully satisfied), and outline future directions which may improve the utility of the current framework or which may expand the application of this framework to other areas in requirements analysis.

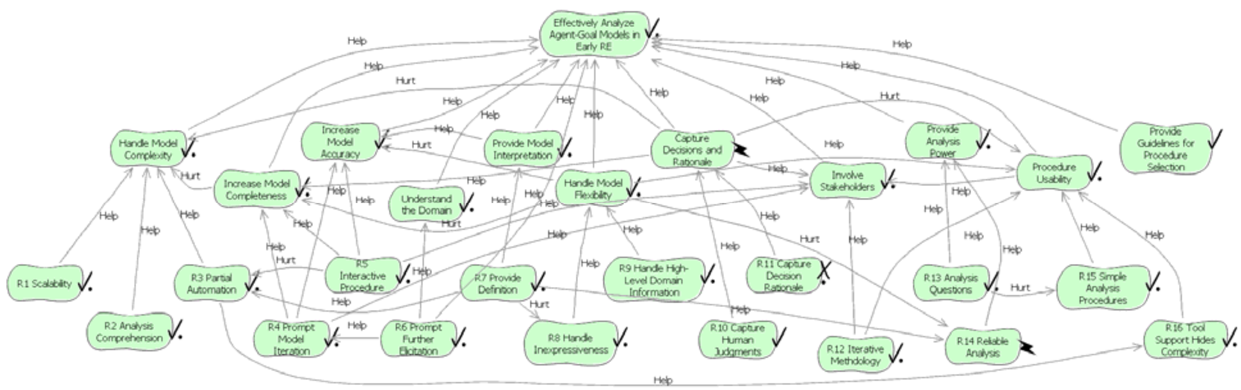


Figure 110 Satisfaction Analysis for the Requirements for analysis of Agent-Goal Models in Early RE based on the Contributions of this Framework

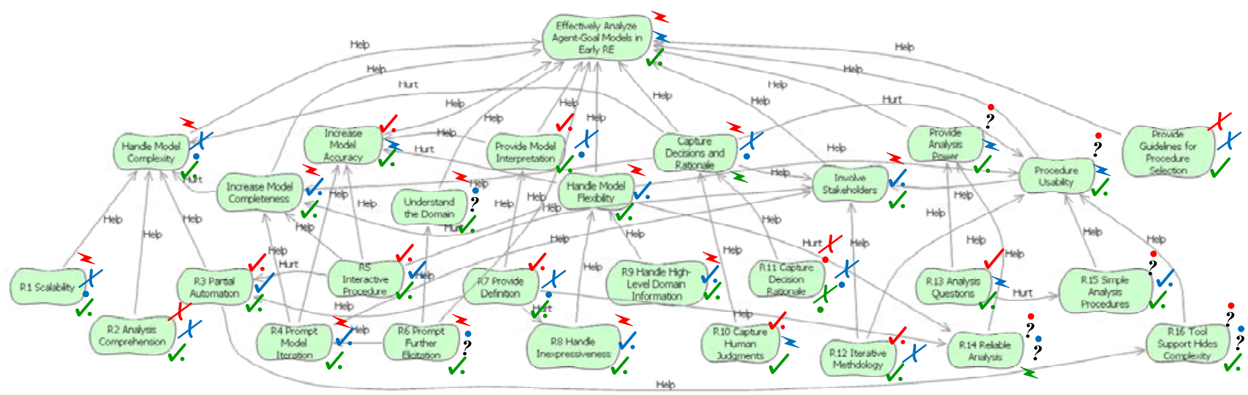


Figure 111: Satisfaction Analysis for the Requirements for analysis of Agent-Goal Models in Early RE based on a Combination of Existing Work including Conflicts and Synergies

13.2 Limitations

By introduction the framework described in this work, we have made significant progress toward effective analysis of early RE agent-goal models, as described by our goals in Chapter 3. However, our approach has several limitations. Several of the limitations have been discussed in the previous section as part of our judgments over the early RE agent-goal model analysis summary model. For example, R14 Analysis Questions was judged to be partially satisfied, as it is always possible to think of more useful analysis questions which could be supported). Further limitations are discussed in the following.

Validation of Procedure Selection Guidelines. Chapter 2 has provided guidelines for the selection of existing goal model analysis procedures by matching procedures to perceived analysis benefits using guiding questions. Although we believe that these guidelines offer a useful first step, they have yet to be validated beyond their application to the examples in Chapter 2, or to the selection of procedures for inclusion in the framework in this thesis. It is possible that our list of perceived benefits of goal model analysis, gathered from the implicit benefits listed in existing work and our own experiences, are not complete. It is also possible that the mapping from benefits to procedures may not apply well in all domains. We welcome future application of these guidelines in the selection of goal model analysis procedures, in the hope that results can be used to iterate over and improve these guidelines.

Framework Requirement Completeness. In Chapter 3, we have used requirements from existing work as well as our own experiences to decompose the perceived challenges of early RE analysis into a set of requirements for such analysis. We believe that these requirements offer an overview of what makes analysis in the early stages of requirements elicitation unique. However, these requirements may not be complete in all settings. In particular, new requirements may arise, or existing requirements may become more or less prominent, depending on the characteristics of a particular domain. For example, in domains with a hierarchical management structure, involving the stakeholders may be less important. In a health-care setting, perhaps early analysis must support an explicit consideration of government regulations. In a domains with a standardized requirements process, perhaps early analysis tool support must be compatible with existing requirements systems. Despite the requirements of specific domains, we argue that the requirements enumerated in this thesis provide an effective generalization of the needs of very early RE analysis.

Goal Modeling Limitations. When we make the decision to support early RE analysis using agent-goal models, we inherit all of the challenges and limitations inherent to this type of modeling. Specifically, we inherit all the limitations of i*. For example, i* can have a learning curve, especially for users who are not experienced modelers, or who are not used to modeling the intentional dimension. Users must have at least a basic knowledge of i* in order to make use of our analysis framework. One of the biggest challenges in i* modeling is the resulting complexity and scalability of the resulting models, as demonstrated by several of our case studies. Although analysis can help to make sense of models, including helping users to determine if models are sufficiently complete and accurate, analysis over complex models can only do so much to ease the cognitive load of complex goal models. If a model is very complex, analysis results over such a model will also be complex, and will be inherently difficult to understand, especially if the user is not familiar with the model or the domain. We have emphasized scoping as a means to ensure that model sizes remain manageable. Future work in the area of agent-goal model scalability, for example, creating modules as in (Morales, Franch, Martínez, Estrada, & Pastor, 2011), could be promising as a point of integration with the approaches in this framework.

Alternative Selection. As highlighted in Chapter 5, it can be difficult to compare the results from multiple satisfaction analysis runs, and use these results to rank alternatives in a systematic or reliable way. The procedures in this framework focus on the evaluation of individual analysis alternatives, and, although multiple results can be viewed using the tool, this work does not provide specific guidance in how to compare the results of multiple analysis alternatives. Given the high-level, incomplete nature of early RE models, it is as problematic to automatically select an alternative using model results as it is to automatically analyze the effects of an alternative in a model. Future work should investigate visualization techniques or methods which help to guide people in comparing and selecting between multiple analysis alternative results.

Generalizability. The procedures and mechanisms introduced in this work have been designed for and applied to the i* framework. We argue that these procedures can generalize relatively easily to very similar frameworks, such as the frameworks described in Chapter 4 (GRL, NFR, Tropos). In fact, the conventions and conversions used in that chapter could help potential users to apply analysis procedures to these slightly differing frameworks.

Applying the procedures and mechanisms from this thesis to other, less similar, goal modeling frameworks such as KAOS, AGORA, or GBRAM would prove more challenging. Interactive analysis as introduced in this work is especially applicable to models containing softgoals and/or contribution links, creating areas of contention in the model, requiring human intervention. If other goal modeling frameworks do not contain such areas, or areas of intervention have a greatly different form (e.g., adding priorities in a matrix such as in AGORA), concepts and algorithms introduced in this work are not easily applicable.

Validation Results. As we have described in Chapter 12, the results of our validation studies are somewhat mixed. Although we have found evidence to support iteration over models and elicitation in the domain as a result of interactive analysis, we have also found cases where this iteration and elicitation does not present itself prominently. We have described hypotheses concerning modeling experience and domain motivation to explain these results. However, several rounds of future study are needed to validate these hypotheses.

In addition, given the design of our examples, case studies, and experiments, it is difficult to attribute the iterative effects of analysis directly to interactive analysis. It is possible that these effects could be caused by a careful examination of the model occurring in ad-hoc analysis, without a systematic procedure, or they could also occur as a result of fully automated analysis, such as in most of the procedures described in Section 2.3. Our Chapter 12 studies showed that systematic analysis resulted in more analysis completeness and consistent interpretations when compared to ad-hoc analysis. As a result, we claim that systematic analysis is more likely to cause elicitation and iteration, as it forces users to consider more of the model, and to consider model constructs in a consistent way.

Some fully automated approaches mention model iteration as a result of analysis (see Section 3.3.7). However, these studies do not explore the condition under which such iteration occurs. Results of the validation studies in our framework indicate that this iteration may only occur with sufficient model and domain expertise and interest. Specifically, in cases where participants did not have extensive domain or model knowledge, we observed users sometimes treated the analysis results as an “oracle”, trusting the results without questioning them. This effect, combined with fully automated analysis, may result in analysis results accepted without question “the model tells me…” without prompting the exploration and iteration needed to improve domain knowledge in early RE. In interactive analysis, users are forced to examine contentious areas of the model via human judgment as part of analysis. We argue that this increases the likelihood of model iteration; however, our studies have shown that inexperienced users sometimes do not change the model, even when their judgments are inconsistent. To this end, we have added support for human judgment consistency checks (Chapter 10). Future work is needed to test what level of procedure automation or interactivity is required under what conditions (modeler experience, domain buy-in) to provoke the most beneficial level of model improvement.

Validation Study Design. Although we retain our claims concerning the iterative and elicitation benefits of interactive analysis, further studies are needed to compare ad-hoc vs. interactive vs. fully automatic analysis. However, we have found through our own iterative process of study design and application that designing such a study, and collecting convincing results, is not trivial. If study design moves towards an action-research approach, a detailed, realistic case study such as the counseling service study, results are more believable due to the realistic complexity of the domain and the involvement of actual stakeholders. However, in this case it is very difficult to apply multiple analysis interventions, especially given the limitations in stakeholder time. We can ask, would it have been possible (if we had developed the research hypotheses concerning the benefits of interactive analysis before the case study application) to apply ad-hoc analysis, then interactive analysis, then fully automated analysis in cooperation with the stakeholders? Although the results from such a study may be interesting, we could not avoid a learning effect, such as in our exploratory study in Chapter 12. Once they have applied ad-hoc analysis, the results for systematic analysis are inevitably altered. If, on the other hand if we lean towards the design of more controlled experiments, such as our exploratory experiment or individual case studies, we have a chance of isolating variables such as the type of analysis, model experience, and domain expertise. However, in order to isolate these variables, we create an artificial situation for modeling and analysis, where users are not sufficiently motivated to understand or answer realistic questions over a domain in which they have some interest. Future studies which further validate the framework must make a careful balance between carefully measuring outcomes and artificiality in case study design.

Validation of the Analysis Framework using the Analysis Framework. As part of the presentation of our analysis framework, we have used the modeling notation of choice, i*, and forward analysis as introduced in this framework as a means to organize and summarize the contributions of this framework. In other words, we have used i* modeling to describe the requirements for early RE agent-goal model analysis, and then forward interactive analysis to describe the ability of our analysis framework to achieve early RE analysis goals, using forward analysis as implemented in this framework. See Figure 110 for an example use of our framework to validate our framework.

We found use of i* modeling and analysis to be very useful for summarizing the achievement of the various components of this work; however, some limitations, such as the lack of granularity of the qualitative labels were revealed, (e.g., PS(Analysis Questions) with some further analysis questions is still judged to be PS, there can always be more analysis questions).

13.3 Future Work

13.3.1 Further Validation

Ideally, a further round of validation would be conducted, testing the methodology and implementation of forward and backward analysis, including new interventions such as human judgment checks and visualizations, in a realistic, complex case study. Such studies could try to test a variety of types of analysis (ad-hoc, interactive, fully automatic) in realistic studies; however, challenges as described in the previous section, balancing measuring the effects of analysis types versus artificial study designs, must be addressed.

13.3.2 Additional Framework Features

Judgment Rationale and Assumptions. We have identified the goal of capturing rationale for decisions as one of the desired requirements for early RE goal model analysis. However, the current framework has not yet addressed this goal. Future work should allow users to enter textual rationale for human judgments. It can also enable users to enter a list of domain assumptions made when giving judgments, similar to the approach in (Maiden et al., 2007), which uses, collects, and attaches satisfaction arguments to i* modeling. These arguments and assumptions can be connected to analysis; i.e., if the assumption is denied, then so is the corresponding intention. Future work should investigate how to relate arguments and analysis values, combining existing approaches.

Varying Levels of Automation. It would be useful to allow users to modify the level of automation. Depending on their confidence in the model (accuracy, completeness), they could select a level of automation along a sliding scale, ranging from judgment in all potentially contentious areas to full automation using set rules to combine evidence, such as in (Amyot et al., 2010). The work described in this thesis likely falls on the far end of the scale away from automation. Although our approach is partially automated, such automation is necessary to facilitate analysis over large and complex models. Any less automation would involve manual propagation, as is done in the forward examples in Section 12.1, which, although possible, is tedious and requires model expertise. Future work should investigate situations where users choose to increase or decrease the level of automation, and how well this facilitates effective RE analysis.

Handling Iteration over Models and Analysis Results. As our analysis framework aims to encourage model iteration, expansions to the framework should handle continuously changing models. More specifically, when a model is changed in some way, this should be reflected on analysis results. There are several ways this could be implemented. A change in a model could prompt an automatic re-evaluation of the model, propagating as far as possible automatically, and then prompting the user if new judgment is needed. However, in an effort to promote model comprehension, it may be better if the user was shown what parts of the analysis results over connected parts of the model were affected by their change, if any. This could be done visually, either in the modeling canvas or in one of the view summarizing analysis results. This would involve deciding whether or not the change would affect analysis results (e.g., renaming an intention may not affect analysis results), and then, for forward analysis, finding the forward model slice from this point. For backward analysis, affected analysis results can be found with both a forward and backward slice, which may in practice highlight most of the model, being less useful. Several student projects supervised by the thesis author have aimed to implement features which handle model changes, with none having completed the full set of implementation tasks. Future work can continue implementation in this area.

Further visualizations. Future work should continue to find ways to enhance goal model analysis application and comprehension through visualization techniques. For example, when changes are made to the model or to human judgments, intentions whose analysis values may be affected by the change could be highlighted for the user. We have focused thus far on intention highlighting, future work can investigate the effectiveness of highlighting links; for example, when they are involved in a conflict or a judgment.

13.3.3 Future Directions

Analysis as a means of Model Validation in the Absence of Stakeholders. Experience in our large-scale counseling service case study has shown that stakeholder time can be scarce, especially when devoted to testing the benefits of new research. In this work, we have investigated use of interactive analysis as a means to improve the model and domain knowledge in early RE, recommending its use in a participatory environment with stakeholders if possible. However, depending of the availability of stakeholder time, it may be difficult to undergo sessions of interactive modeling and analysis. We have described how to use analysis as a means to check the sanity of a model, especially after it has been first drawn. These ideas could be combined to investigate how analysis, interactive or otherwise, could be used as a means of model validation in the absence, or near absence, of stakeholders. This has been addressed, to a certain extent, but our investigation into the effects of interactive analysis on further elicitation questions, but these ideas could be taken farther, looking at ways to use analysis to find key unknown areas over which to target stakeholder time.

From Early to Late RE. We have described techniques which focus on analysis over early RE agent-goal models, supporting high-level concepts without yet going into technical detail. Future work should guide users in moving from this type of model, and the type of analysis introduced in this work, into more detailed, more carefully scoped RE models. Such are the models introduced and used in many of the existing goal model analysis approaches, requiring detailed information such as probability, priority, or temporal ordering. How do users move effectively from the very early RE modeling and analysis used in this work to more detailed models used in other goal model analysis approaches?

Although we have focused the techniques in this thesis on early RE analysis, where there is an absence for quantitative or formal domain information, it may be possible to elicit partial, more specific, domain information as part of even the early RE process. There is a question of whether or not collecting these specific metrics may distract from understanding the big picture, as is aimed for with high-level early RE models. However, if detailed information arises, it could be collected, potentially attached to the model, and left for further investigation and expansion in later RE stages, after high-level alternatives have been selected rounds of scoping have occurred.

Work towards simultaneously using early qualitative and later quantitative analysis has been introduced. These approaches are useful in that they apply more specific, likely more reliable, analysis but do not require that the detailed information required for this analysis be complete over the model. For example, in (Barone et al., 2011) analysis over for Business Intelligence Models can be qualitative over less well specified areas of the model, and quantitative, potentially using domain-specific equations, in other, more specified areas. Different types of analysis results are mapped together, facilitating propagation between the two. The qualitative analysis introduced in this thesis could fit well into the qualitative analysis described within this approach. Similar work in (Pourshahid et al., 2011) mixes domain specific equations with generic quantitative propagation in GRL.

We have provided a high-level view of a process which moves from early to later RE analysis in Figure 13, repeated below. The Framework in this thesis focuses solely on the top box. Future work should expand on this framework to describe how to move between the first and second boxes in the Figure.

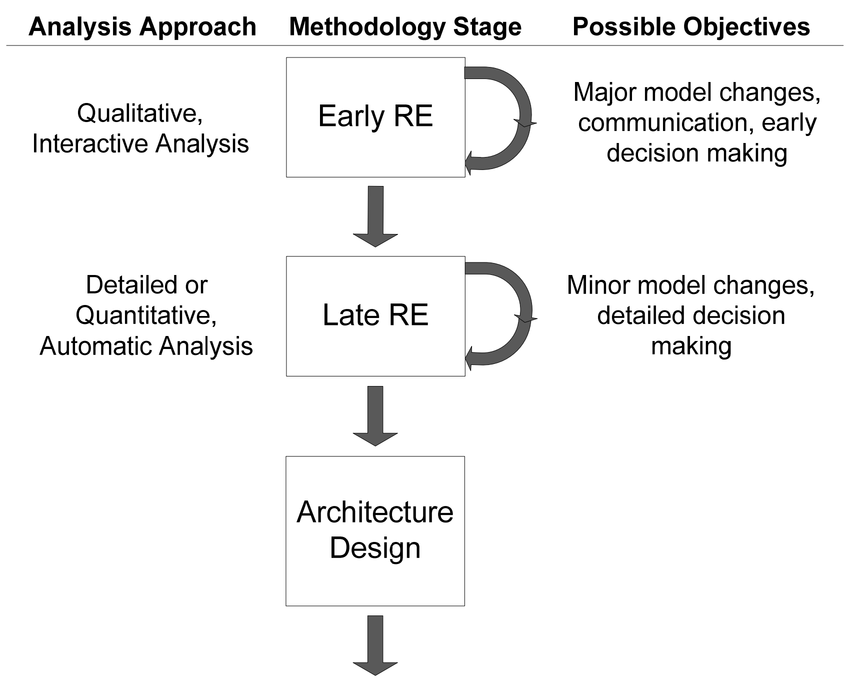


Figure 112: Example development process using both qualitative and quantitative goal model analysis

Confidence in Analysis Results. The comparison study in Chapter 4 has focused on the reliability of analysis results over early RE models, showing that results can be inconsistent when the models contain many "social" concepts. We have highlighted a tradeoff between inexpressiveness and effective early RE modeling and the precise and detailed information needed for more accurate analysis results. Future work can aim to measure the perceived confidence in analysis results based on several factors such as: confidence in the sources of the model, the structure of the model (e.g., how many softgoals), the length of propagation paths, the sources of initial evaluation values, and the means of propagation (e.g., qualitative through propagation links or quantitative using domain-specific formula). Confidence could be over the entire results or over analysis values over individual intentions. Such confidence measures associated with analysis values can help to guide users in whether or not the analysis results should be used as a heuristic only, or can be more trusted, closer to a simulation over concrete domain measures.

Such confidence measures would be especially useful in an approach which leads from early, interactive, qualitative analysis to later, quantitative automatic analysis. Earlier, less detailed analysis would likely receive lower confidence values than later more detailed analysis. Hybrid models leading from early to later RE may have a mix of confidence levels, depending on the evidence sources and means of propagation.

Bibliography

Amyot, D. (2003). Introduction to the User Requirements Notation: learning by example. Computer Networks, 42(3), 285-301. Elsevier.

Amyot, D., Ghanavati, S., Horkoff, J., Mussbacher, G., Peyton, L., & Yu, E. (2010). Evaluating goal models within the goal-oriented requirement language. International Journal of Intelligent Systems, 25(8), 841-877.

Anderson, R. (2001). Why information security is hard - an economic perspective. Seventeenth Annual Computer Security Applications Conference, 358-365. IEEE Comput. Soc.

Anton, A. I., & Potts, C. (1998). The Use of Goals to Surface Requirements for Evolving Systems. Proceedings of the 20th international conference on Software engineering (pp. 157-166).

Asnar, Y., & Giorgini, P. (2006). Modelling Risk and Identifying Countermeasure in Organizations. (J. Lopez, Ed.)Critical Information Infrastructures Security, 4347(July), 55-66. Springer.

Asnar, Y., Bryl, V., & Giorgini, P. (2007). Using Risk Analysis to Evaluate Design Alternatives. AgentOriented Software Engineering VII, 140-155. Springer.

Barone, D., Jian, L., Amyot, D., & Mylopoulos, J. (2011). Composite Indicators for Business Intelligence. ER’11: Proceedings of the 30th International Conference on Conceptual Modeling.

Barone, D., Yu, E., Won, J., Jiang, L., & Mylopoulos, J. (2010). Enterprise Modeling for Business Intelligence. Ifip International Federation For Information Processing, 22(1), 31-45. IBM Press.

Bresciani, P., Perini, A., Giorgini, P., Giunchiglia, F., & Mylopoulos, J. (2004). Tropos: An Agent-Oriented Software Development Methodology. Autonomous Agents and MultiAgent Systems, 8(3), 203-236. Springer.

Bruni, R., & Sassano, A. (2001). Restoring Satisfiability or Maintaining Unsatisfiability by finding small Unsatisfiable Subformulae. Electronic Notes in Discrete Mathematics, 9, 162-173.

Bryl, V., Dalpiaz, F., Ferrario, R., Mattioli, A., & Villafiorita, A. (2009b). Evaluating procedural alternatives: a case study in e-voting. Electronic Government an International Journal, 6(2), 213-231. Inderscience.

Bryl, V., Giorgini, P., & Mylopoulos, J. (2006a). Designing cooperative IS: Exploring and evaluating alternatives. On the Move to Meaningful Internet Systems 2006 CoopIS DOA GADA and ODBASE, 533–550. Springer.

Bryl, V., Giorgini, P., & Mylopoulos, J. (2009a). Supporting Requirements Analysis in Tropos: A Planning-Based Approach. In A. Ghose, G. Governatori, & R. Sadananda (Eds.), Agent Computing and Multi-Agent Systems (Vol. 5044, pp. 243-254). Springer Berlin Heidelberg.

Bryl, V., Massacci, F., Mylopoulos, J., & Zannone, N. (2006b). Designing Security Requirements Models Through Planning. In E. Dubois & K. Pohl (Eds.), Advanced Information Systems Engineering (Vol. 4001, pp. 33-47). Springer Berlin / Heidelberg.

Cabot, J., Easterbrook, S, Horkoff, J., Lessard, L., Liaskos, S., & Mazon, J.-N. (2009). Integrating sustainability in decision-making processes: A modelling strategy. 2009 31st International Conference on Software Engineering Companion Volume, 207-210.

Cares, C., & Franch, X. (2008). iStarML: an XML-based Interchange Format for i* Models. Third International iStar Workshop, 1-3.

Cares, C., & Franch, X. (2011). A Metamodelling Approach for i* Model Translations. In H. Mouratidis & C. Rolland (Eds.), Advanced Information Systems Engineering (Vol. 6741, pp. 337-351). Springer Berlin / Heidelberg.

Cares, C., Franch, X., Colomer, D., & López, L. (2011). Tool Interoperability using iStarML. iStar Tool Fair, 5th Internation i* Workshop, Co-located with RE’11.

Castro, J., Kolp, M., & Mylopoulos, J. (2001). A Requirements-Driven Development Methodology. Advanced Information Systems Engineering 13th International Conference CAiSE 2001Interlaken Switzerland, 108-123.

Checkland, P. (2000). Soft systems methodology: a thirty year retrospective. Systems Research and Behavioral Science, 17(S1), S11-S58.

Chung, L., Nixon, B. A., Yu, E., & Mylopoulos, J. (2000). Non-Functional Requirements in Software Engineering. Conceptual Modeling Foundations and Applications (Vol. 5, pp. 363–379). Kluwer Academic Publishers.

Clarke, E. M., Grumberg, O., & Peled, D. A. (2000). Model Checking. (E. A. Emerson & A. P. Sistla, Eds.) (p. 314). MIT Press.

Dano, B., Briand, H., & Barbier, F. (1997). A use case driven requirements engineering process. Requirements Engineering, 2(2), 79-91. Springer London.

Dardenne, A., Lamsweerde, VAN, & Fickas, S. (1993). Goal-directed requirements acquisition. Science of Computer Programming, 20(1-2), 3-50. Elsevier Science B.V.

Easterbrook, S., Yu, E., Aranda, J., Fan, Y., Horkoff, J., Leica, M., & Qadir, R. A. (2005). Do Viewpoints Lead to Better Conceptual Models? An Exploratory Case Study. 13th IEEE International Conference on Requirements Engineering RE05, 199-208.

Easterbrook, S., Yu, E., Aranda, J., Horkoff, J., Strohmaier, M., Fan, Y., Leica, M., et al. (2011). Analyzing Requirements for Online Presence Kids Help Phone. iStar Showcase.

Elahi, G., & Yu, E. (2007). A Goal Oriented Approach for Modeling and Analyzing Security Trade-Offs. Security, 4801(7), 375-390. Springer Berlin Heidelberg.

Elahi, G., & Yu, E. (2011). Requirements trade-offs analysis in the absence of quantitative measures: a heuristic method. Proceedings of the 2011 ACM Symposium on Applied Computing (pp. 651-658). New York, NY, USA: ACM.

Elahi, G., Yu, E. S. K., & Annosi, M. C. (2008). Modeling Knowledge Transfer in a Software Maintenance Organization - An Experience Report and Critical Analysis. In J. Stirna &

A. Persson (Eds.), PoEM (Vol. 15, pp. 15-29). Springer. Retrieved from http://dblp.uni-trier.de/db/conf/ifip8-1/poem2008.html#ElahiYA08

Ernst, N. A., Mylopoulos, J., Borgida, A., & Jureta, I. J. (2010). Reasoning with Optional and Preferred Requirements. ER, 6412, 118-131. Springer Berlin / Heidelberg.

Ernst, N., Yu, Y., & Mylopoulos, J. (2006). Visualizing non-functional requirements. 2006 First International Workshop on Requirements Engineering Visualization REV06 RE06 Workshop, 2-2.

Feather, M., & Cornford, S. (2003). Quantitative Risk-based Requirements Reasoning. (J Mylopoulos & P. Lycopoulos, Eds.)Requirements Engineering Journal, 8(4), 248-265. Springer.

Fifth International i* Workshop. (2011). Retrieved from http://www.cin.ufpe.br/~istar11

Forbus, K. (1997). Qualitative Reasoning. In A. B. Tucker (Ed.), CRC Handbook of Computer Science and Engineering.

Franch, X. (2006). On the Quantitative Analysis of Agent-Oriented Models. Advanced Information Systems Engineering, LNCS vol. 4001 (pp. 495-509). Springer.

Franch, X. (2009). A method for the definition of metrics over i* models. In J. Gordijn & Roel Wieringa (Eds.), Advanced Information Systems Engineering Proceedings 21st International Conference CAiSE 2009 (Vol. 5565, pp. 201-215). Springer - Verlag Berlin Heidelberg.

Franch, X., & Maiden, N. A. M. (2003). Modelling Component Dependencies to Inform Their Selection. In H. Erdogmus & T. Weng (Eds.), COTS-Based Software Systems (Vol. 2580, pp. 81-91). Springer Berlin / Heidelberg.

Franch, X., Grau, G., & Quer, C. (2004). A framework for the definition of metrics for actor-dependency models. Proceedings 12th IEEE International Requirements Engineering Conference 2004 (pp. 348-349). IEEE Computer Society.

Fuxman, A., Liu, L., Pistore, M., Roveri, M., & Mylopoulos, J. (2003). Specifying and analyzing early requirements: some experimental results. Proceedings. 11th IEEE International Requirements Engineering Conference (RE'03) (pp. 105-114). IEEE.

Fuxman, A., Pistore, M., Mylopoulos, John, & Traverso, P. (2001). Model checking early requirements specifications in Tropos. Proceedings Fifth IEEE International Symposium on Requirements Engineering (pp. 174-181). IEEE Comput. Soc.

Gans, G., Jarke, M., Kethers, S., & Lakemeyer, G. (2003b). Continuous requirements management for organisation networks: a (dis)trust-based approach. Requirements Engineering, 8(1), 4-22.

Gans, G., Jarke, M., Kethers, S., Lakemeyer, G., Ellrich, L., Funken, C., et al. (2001). Requirements modeling for organization networks: a (dis-)trust-based approach. Proceedings RE01 5th International Symposium on Requirements Engineering. 2001 (Vol. August, pp. 154-163).

Gans, G., Jarke, M., Lakemeyer, G., & Schmitz, D. (2003a). Deliberation in a modeling and simulation environment for inter-organizational networks. Proc Of the 15th Int Conf On Advanced Information System Engineering CAiSE03 Klagenfurt Austria June (Vol. 2681, pp. 242-257). Springer-Verlag.

Gans, G., Jarke, M., Lakemeyer, G., & Schmitz, D. (2005). Deliberation in a metadata-based modeling and simulation environment for inter-organizational networks. Information Systems Journal, 30(7), 587-607.

Gans, G., Jarke, M., Lakemeyer, G., & Vits, T. (2002). SNet: A modeling and simulation environment for agent networks based on i* and ConGolog. Lecture Notes in Computer Science (Vol. 3248, pp. 328-343).

Gans, G., Schmitz, D., Arzdorf, T., Jarke, M., & Lakemeyer, G. (2005). SNet Reloaded: Roles, Monitoring and Agent Evolution. In P. Bresciani, P. Giorgini, B. Henderson-Sellers, G. Low, & M. Winikoff (Eds.), Agent-Oriented Information Systems II (Vol. 3508, p. 431). Springer Berlin / Heidelberg.

Gelder, T. V. (2009). What is argument mapping? Retrieved from http://timvangelder.com/2009/02/17/what-is-argument-mapping/.

Giorgini, P., Massacci, F., Mylopoulos, John, & Zannone, N. (2004c). Requirements Engineering Meets Trust Management, Model, Methodology, and Reasoning. Lecture Notes in Computer Science (Vol. 2995, pp. 176-190). Springer-Verlag. Retrieved from

Giorgini, P., Mylopoulos, J., & Sebastiani, R. (2004b). Simple and Minimum-Cost Satisfiability for Goal Models. 16th Conference On Advanced Information Systems Engineering (CAiSE*04).

Giorgini, P., Mylopoulos, J., & Sebastiani, R. (2005). Goal-oriented requirements analysis and reasoning in the Tropos methodology. Engineering Applications of Artificial Intelligence, 18(2), 159-171. Elsevier.

Giorgini, P., Mylopoulos, J., Nicchiarelli, E., & Sebastiani, R. (2002). Reasoning with Goal Models. ER '02: Proceedings of the 21st International Conference on Conceptual Modeling (pp. 167-181). London, UK: Springer-Verlag.

Giorgini, P., Mylopoulos, J., Nicchiarelli, E., & Sebastiani, R. (2004a). Formal Reasoning Techniques for Goal Models. Journal on Data Semantics, 2800/2003(1), 1-20. Springer.

Gordijn, J., Petit, M., & Wieringa, R. (2006). Understanding Business Strategies of Networked Value Constellations Using Goal- and Value Modeling. Proceedings of the IEEE International Conference on Requirements Engineering (pp. 126-135).

Gottwald, S. (2000). Many-Valued Logic. Retrieved from http://plato.stanford.edu/entries/logic-manyvalued/

Grau, G., & Franch, X. (2007). A Goal-Oriented Approach for the Generation and Evaluation of Alternative Architectures. In Oquendo F, vol, 139-155. Springer Verlag.

Grau, G., Franch, X., & Ávila, S. (2006). J-PRiM: A Java Tool for a Process Reengineering i* Methodology. 14th IEEE International Conference on Requirements Engineering RE06 (p. 359–360). IEEE Computer Society Press.

Grau, G., Franch, X., & Maiden, N. A. M. (2005). A Goal-Based Round-Trip Method for System Development. Proceedings of the 11th International Conference on Requirements Engineering: Foundations for Software Quality (REFSQ'05).

Grau, G., Franch, X., & Maiden, N. A. M. (2008). PRiM: An i*-based process reengineering method for information systems specification. Inf. Softw. Technol., 50(1-2), 76-100. Newton, MA, USA: Butterworth-Heinemann.

Grau, G., Franch, X., Mayol, E., Ayala, C., & Cares, C. (2005). RiSD: A Methodology for Building i* Strategic Dependency Models. 17th International Conference on Software Engineering and Knowledge Engineering SEKE05 (pp. 259-266). Citeseer.

Grau, G., Horkoff, J., Schmitz, D., Abdulhadi, Samer, & Yu, E. S. K. (2008). Fostering Investigation, Collaboration, and Evaluation: the i* Wiki Experience. iStar (pp. 33-36).

Grau, G., Horkoff, J., Yu, E., & Abdulhadi, S. (2010). IStar Guide. Retrieved from http://istar.rwth--aachen.de/tiki--index.php?page_re

Group, O. M. (2009). OMG Model Driven Architecture. Retrieved from http://www.omg.org/mda/.

GR-Tool. (2010). GR-Tool. Retrieved from http://troposproject.org/tools/grtool/.

Horkoff, J. (2006). An Evaluation Algorithm for the i* Framework. University of Toronto.

Horkoff, J. (2008a). Analysis of Goal- and Agent-Oriented Models, Depth Oral Report.

Horkoff, J. (2008b). CSC2108 Course Project: Qualitative, Interactive, Backward Analysis of i* Models.

Horkoff, J. (2009). Instruction Manual: A Qualitative, Interactive Evaluation Procedure for the i* Framework. Retrieved from http://www.cs.utoronto.ca/~jenhork/PhDThesis/ForwardProcedureDescription_ExploratoryExperiment.pdf

Horkoff, J. (2011a). Using i* Modeling for the Analysis of Strategy Documents: Course Project for CS2507 Conceptual Modeling, Spring 2007. DCS Technical Report, CSRG-613. ftp://ftp.cs.toronto.edu/csrg-technical-reports/613/

Horkoff, J. (2011b). Applying the i* Framework, Scenarios, and Prioritization Methods to a Requirements Analysis for Kids Help Phone: Course Project for CS2106 Requirements Engineering, Fall 2005. DCS Technical Report, CSRG-612. ftp://ftp.cs.toronto.edu/csrg-technical-reports/612/

Horkoff, J., & Yu, E. (2008). Qualitative, Interactive, Backward Analysis of i* Models. 3rd International i* Workshop (pp. 4-46). CEUR-WS.org.

Horkoff, J., & Yu, E. (2009a). Evaluating Goal Achievement in Enterprise Modeling – An Interactive Procedure and Experiences. The Practice of Enterprise Modeling (pp. 145-160). Springer.

Horkoff, J., & Yu, E. (2009b). A Qualitative, Interactive Evaluation Procedure for Goal- and Agent-Oriented Models. CAiSE'09 Forum, Vol-453 (pp. 19-24). CEUR-WS.org.

Horkoff, J., & Yu, E. (2010a). Finding Solutions in Goal Models: An Interactive Backward Reasoning Approach. Conceptual Modeling (ER 2010), 29th International Conference on Conceptual Modeling, Vol. 6412, p. 59. Springer-Verlag New York Inc.

Horkoff, J., & Yu, E. (2010b). Interactive Analysis of Agent-Goal Models in Enterprise Modeling. International Journal of Information System Modeling and Design (IJISMD). IGI Global.

Horkoff, J., & Yu, E. (2010c). Visualizations to support interactive goal model analysis. Requirements Engineering Visualization REV 2010 Fifth International Workshop on (Vol. 70, p. 1–10). IEEE.

Horkoff, J., & Yu, E. (2010d). Interactive Goal Model Analysis Applied - Systematic Procedures versus Ad hoc Analysis. The Practice of Enterprise Modeling, 3rd IFIP WG8.1 (PoEM'10).

Horkoff, J., & Yu, E. (2011a). Analyzing goal models: different approaches and how to choose among them. Proceedings of the 2011 ACM Symposium on Applied Computing (pp. 675-682). New York, NY, USA: ACM.

Horkoff, J., & Yu, E. (2011b). Comparison and Evaluation of Goal-Oriented Satisfaction Analysis Techniques. Requirements Engineering (REJ) (conditionally accepted).

Horkoff, J., & Yu, E. (2011c). Detecting Judgment Inconsistencies to Encourage Model Iteration in Interactive i* Analysis. Proceedings of the 5th International i* Workshop (pp. 2-7).

Horkoff, J., Yu, E. (2011d). Modeling and Analyzing Technology Strategies. DCS Techncial Report, CSRG-614. ftp://ftp.cs.toronto.edu/csrg-technical-reports/614/

Horkoff, J., Elahi, G., Abdulhadi, Samer, & Yu, E. (2008). Reflective Analysis of the Syntax and Semantics of the i* Framework. Lecture Notes in Computer Science, 5232, 249-260.

Horkoff, J., Yu, E., & Liu, L. (2006). Analyzing trust in technology strategies. Proceedings of the 2006 International Conference on Privacy Security and Trust Bridge the Gap Between PST Technologies and Business Services PST 06 (p. 1). ACM Press.

Horkoff, J., Yu, Y., & Yu, E. (2011). OpenOME: An Open-source Goal and Agent-Oriented Model Drawing and Analysis Tool. iStar Tool Fair: Proceedings of the 5th International i* Workshop (pp. 154-156).

i* Readings Roadmap. (2011). Retrieved from http://www.cs.utoronto.ca/km/istar/#roadmap

i* Wiki. (2010). No Title.

International Telecommunication Union, Telelcommuniction Standardization Sector, S. G. 17. (2008). Z.151: User Requirements Notaiton (URN). Retrieved from http://jucmnav.softwareengineering.ca/twiki/bin/view/UCM/DraftZ151Standard.

ITU-T. (2008). Recommendation Z.151 (11/08) User Requirements Notation (URN) – Language definition. Geneva, Switzerland.

Jackson, M. (2001). Problem Frames. ACM Press Books.

JUCMNav. (2011). jUCMNav - Eclipse plugin for the User Requirements Notation. Retrieved from http://jucmnav.softwareengineering.ca/ucm/bin/view/ProjetSEG/WebHome

Jureta, I. J., Borgida, A., Ernst, N. A., & Mylopoulos, J. (2010). Techne: Towards a New Generation of Requirements Modeling Languages with Goals, Preferences, and Inconsistency Handling. 2010 18th IEEE International Requirements Engineering Conference (pp. 115-124).

Jureta, I., Mylopoulos, John, & Faulkner, S. (2008). Revisiting the Core Ontology and Problem in Requirements Engineering. 2008 16th IEEE International Requirements Engineering Conference, 71-80. IEEE.

Kaiya, H., Horai, H., & Saeki, M. (2002). AGORA: attributed goal-oriented requirements analysis method. Requirements Engineering 2002 Proceedings IEEE Joint International Conference on (pp. 13-22). IEEE.

Kaplan, R. S., & Norton, D. P. (1992). The Balanced Scorecard – Measures that Drive Performance The Balanced Scorecard — Measures. Harvard Business Review, 70(1), 71-79.

Kaplan, R. S., & Norton, D. P. (2001). The Strategy-Focused Organization: How Balanced Scorecard Companies thrive in the new Business Environment (p. x, 400 p.). Harvard Business School Press.

Lamsweerde, van (2001). Goal-oriented requirements engineering: a guided tour. Proceedings Fifth IEEE International Symposium on Requirements Engineering, 249(August), 249-262. IEEE Comput. Soc.

Lamsweerde, van (2004). Goal-oriented requirements engineering: A roundtrip from research to practice. Proceedings of the 12th IEEE International Requirements Engineering Conference.

Lamsweerde, van (2009). Reasoning About Alternative Requirements Options. In A. Borgida, V. K. Chaudhri, P. Giorgini, & E. S. K. Yu (Eds.), Conceptual Modeling Foundations and Applications (Vol. 5600, pp. 380-397). Springer.

Lamsweerde, van, Darimont, R., & Letier, E. (1998). Managing conflicts in goal-driven requirements engineering. IEEE Transactions on Software Engineering, 24(11), 908-926. IEEE.

Laue, R., & Storch, A. (2011). Adding Functionality to openOME for Everyone. iStar Tool Fair, 5th Internation i* Workshop, Co-located with RE'11.

Leite, D. P., Sampaio, J. C., Yu, Y., Liu, L., Yu, E., & Mylopoulos, John. (2005). Quality-Based Software Reuse. CAISE, 3520, 535-550.

Letier, E, & Lamsweerde, van (2002). Agent-based tactics for goal-oriented requirements elaboration. Proceedings of the 24th international conference on Software engineering ICSE 02, (May), 83. ACM Press.

Letier, E, & Lamsweerde, van (2004). Reasoning about partial goal satisfaction for requirements and design engineering. ACM SIGSOFT Software Engineering Notes, 29(6), 53-62. ACM.

Liaskos, S., McIlraith, S. A., Sohrabi, S., & Mylopoulos, J. (2011). Representing and reasoning about preferences in requirements engineering. Requirements Engineering (REJ), 16(3), 227-249. Springer London.

Liu, L., & Yu, E. (2004). Designing information systems in social context: a goal and scenario modelling approach. Information Systems Journal, 29(2), 187-203.

Liu, L., Yu, E., & Mylopoulos, John. (2003). Security and Privacy Requirements Analysis within a Social Setting. RE 03 Proceedings of the 11th IEEE International Conference on Requirements Engineering (Vol. 3, pp. 151–161). IEEE Computer Society.

Mahajan, Y. S., Fu, Z., & Malik, S. (2004). Zchaff2004: An Efficient SAT solver. Proc. Seventh International Conf. on Theory and Applications of Satisfiability Testing (SAT'04) (pp. 360-375).

Maiden, N. A. M., Jones, S., Manning, S., Greenwood, J., & Renou, L. (2004). Model-Driven Requirements Engineering : Synchronising Models in an Air Traffic Management Case

Study. In A. Persson & J. Stirna (Eds.), Intl Conf Advanced Information Systems Eng CAiSE (Vol. 3084, pp. 3-21). Springer Berlin/Heidelberg.

Maiden, N. A. M., Lockerbie, J., Randall, D., Jones, S., & Bush, D. (2007). Using Satisfaction Arguments to Enhance i* Modelling of an Air Traffic Management System. Requirements Engineering, IEEE International Conference on, 49-52. IEEE Computer Society.

Moody, D. L., Heymans, P., & Matulevicius, R. (2009). Improving the Effectiveness of Visual Representations in Requirements Engineering: An Evaluation of i* Visual Syntax. (P. Heymans & R. Matulevicius, Eds.)2009 17th IEEE International Requirements Engineering Conference, 0, 171-180. IEEE.

Morales, E., Franch, X., Martínez, A., Estrada, H., & Pastor, O. (2011). Technology Representation in i* Modules. Fifth International iStar Workshop (iStar'11).

Mylopoulos, John, Borgida, A., Jarke, M., & Koubarakis, M. (1990). Telos: representing knowledge about information systems. ACM Transactions on Information Systems, 8(4), 325-362. ACM.

Niu, N., & Easterbrook, Steve. (2006). Managing Terminological Interference in Goal Models with Repertory Grid. 14th IEEE International Requirements Engineering Conference RE06, 303-306. IEEE.

OME, Organization Modelling Environment. (2008). . Retrieved from http://www.cs.toronto.edu/km/ome/

OpenOME Trac Wiki. (2011). . Retrieved from https://se.cs.toronto.edu/trac/ome/

OpenOME, an open-source requirements engineering tool. (2010). . Retrieved from http://www.cs.toronto.edu/km/openome/

OpenOME, Sourceforge. (2011). . Retrieved from http://sourceforge.net/projects/openome/

Osterwalder, A. (2004). The Business Model Ontology - A proposition in a desing science approach. Business. l'Université de Lausanne.

Persson, A., & Stirna, J. (2002). An explorative study into the influence of business goals on the practical use of Enterprise Modelling methods and tools. In G. Harindranath, W. Wojtkowski Gregory, J. Zupancic, D. Rosenberg, W. Wojtkowski, S. Wrycza, & A. A. Sillince John (Eds.), New Perspectives on Information Systems Development Theory Methods and Practice (p. 13). Kluwer Academic/Plenum Publisher.

Pourshahid, A, Amyot, D., Peyton, L., Ghanavati, S., Chen, P., Weiss, M., & Forster, A. J. (2008). Toward an Integrated User Requirements Notation Framework and Tool forBusiness Process Management. eTechnologies 2008 International MCETECH Conference on (pp. 3-15). IEEE.

Pourshahid, A, Richards, G., & Amyot, D. (2011). Toward a Goal-Oriented, Business Intelligence Decision-Making Framework. In W. Aalst, John Mylopoulos, N. M. Sadeh, M. J. Shaw, C. Szyperski, G. Babin, K. Stanoevska-Slabeva, et al. (Eds.), ETechnologies Transformation in a Connected World (Vol. 78, pp. 100-115). Springer Berlin Heidelberg.

Power, D. J. (2007). A Brief History of Decision Support Systems. Decision Support Systems, 2008(1969), 1-23. DSS Resources. Retrieved from http://dssresources.com/history/dsshistory.html

Robertson, S., & Robertson, J. (2006). Mastering the Requirements Process (2nd Edition). Addison-Wesley Professional.

Robinson, W. N., Pawlowski, S. D., & Volkov, V. (2003). Requirements interaction management. ACM Computing Surveys, 35(2), 132-190. ACM.

Rohleder, C. (2008). Visualizing the Impact of Non-Functional Requirements on Variants: A Case Study. 2008 Third International Workshop on Requirements Engineering Visualization REV08 RE08 Workshop (pp. 11-20). IEEE Computer Society.

Rolland, C., Stirna, J., Prekas, N., Loucopoulos, P., Persson, A., & Grosz, G. (2000). Evaluating a Pattern Approach as an Aid for the Development of Organisational Knowledge: An Empirical Study. Proceedings of the 12th International Conference on Advanced Information Systems Engineering (pp. 176-191). London, UK: Springer-Verlag.

Roy, J.-F., Kealey, J., & Amyot, D. (2006). Towards Integrated Tool Support for the User Requirements Notation. In R. Gotzhein & R. Reed (Eds.), System Analysis and Modeling: Language Profiles (Vol. 4320, pp. 198-215). Springer Berlin / Heidelberg.

Samavi, R., Yu, E., & Topaloglou, T. (2008). Strategic reasoning about business models: a conceptual modeling approach. Information Systems and eBusiness Management, 7(2), 171-198.

Sawyer, P., Bencomo, N., Hughes, D., Grace, P., Goldsby, H. J., & Cheng, B. H. C. (2007). Visualizing the Analysis of Dynamically Adaptive Systems Using. Requirements Engineering, (Rev).

Seaman, C. B. (1999). Qualitative methods in empirical studies of software engineering. IEEE Transactions on Software Engineering, 25(4), 557-572. IEEE Press.

Sebastiani, R., Giorgini, P., & Mylopoulos, John. (2004). Simple and Minimum-Cost Satisfiability for Goal Models. Lecture Notes in Computer Science, 3084/2004, 20-35. Citeseer.

Sen, A., & Jain, S. K. (2007). A Visualization Technique for Agent Based Goal Refinement to Elicit Soft Goals in Goal Oriented Requirements Engineering. 2007 Second International Workshop on Requirements Engineering Visualization REV07 RE07 Workshop.

Simon, H. A. (1969). The Science of the Artificial. Neurosurgery (Vol. 56, p. 123). MIT Press.

Smith, B. C. (2004). Stepping Up: Information Practice in the 21st Century. Retrieved from http://www.fis.utoronto.ca/images/documents/about/fisacademicplan.pdf

Spiekermann, S., Dickinson, I., Günther, O., & Reynolds, D. (2003). User agents in E-commerce environments: industry vs. consumer perspectives on data exchange. Proceedings of the 15th international conference on Advanced information systems engineering (pp. 696-710). Berlin, Heidelberg: Springer-Verlag.

Stirna, J., & Persson, A. (2007). Ten Years Plus with EKD: Reflections from Using an Enterprise Modeling Method in Practice. In B. Pernici & J. A. Gulla (Eds.), Proceedings of the 11th

International Workshop on Exploring Modeling Methods in Systems Analysis and Design (EMMSAD'07) (pp. 99-108). CEUR Workshop Proceedings.

Stirna, J., Persson, A., & Sandkuhl, K. (2007). Participative Enterprise Modeling: Experiences and Recommendations. In G. Krogstie, John and Opdahl, Andreas and Sindre (Ed.), Advanced Information Systems Engineering (Vol. 4495, pp. 546-560). Springer Berlin / Heidelberg.

Strohmaier, M., Horkoff, J., Yu, E., Aranda, J., & Easterbrook, S. (2008). Can Patterns Improve i* Modeling? Two Exploratory Studies. Proceedings of the 14th International Working Conference on Requirements Engineering Foundation for Software Quality (pp. 153-167). Springer.

Strohmaier, M., Yu, E., Horkoff, J., Aranda, J., & Easterbrook, S. (2007). Analyzing Knowledge Transfer Effectiveness – An Agent-Oriented Modeling Approach. Proceedings of the 40th Annual Hawaii International Conference on System Sciences (p. 188b). IEEE Computer Society.

Tanabe, D., Uno, K., Akemine, K., Yoshikawa, T., Kaiya, H., & Saeki, M. (2008). Supporting Requirements Change Management in Goal Oriented Analysis. 2008 16th IEEE International Requirements Engineering Conference, 3-12. IEEE.

The Business Motivation Model Business Governance in a Volatile World. (2005).

The National Security Strategy of the United States of America. (2006). Retrieved from http://www.whitehouse.gov/nsc/nss/2006

Wang, X., & Lespérance, Y. (2001). Agent-oriented requirements engineering using ConGolog and i. Agentoriented Information Systems 2001 Proceedings of AOIs2001 Third International BIconference Workshop 28 May 2001 Montreal AT Agents 2001 4 June 200 Interlaken AT Caise01 (p. 59).

Wang, Y., McIlraith, S. A., Yu, Y., & Mylopoulos, John. (2007). An automated approach to monitoring and diagnosing requirements. Proceedings of the twentysecond IEEEACM international conference on Automated software engineering ASE 07, 293. ACM Press.

Yu, E. (1995). Modelling Strategic Relationships for Process Reengineering. Computer Science. University of Toronto.

Yu, E. (1997). Towards modelling and reasoning support for early-phase requirements engineering. Proceedings of ISRE 97 3rd IEEE International Symposium on Requirements Engineering, 97, 226-235. IEEE Comput. Soc. Press.

Yu, E., & Cysneiros, L. (2003). Designing for Privacy in a Multi-agent World. Trust Reputation and Security Theories and Practice, 259-269. Springer.

Zave, P., & Jackson, M. (1997). Four dark corners of requirements engineering. ACM Transactions on Software Engineering and Methodology, 6(1), 1-30. ACM.

Zhang, J., Li, S., & Shen, S. (2006). Extracting Minimum Unsatisfiable Cores with a Greedy Genetic Algorithm. Australian Conference on Artificial Intelligence (pp. 847-856).

Appendices

Appendix A i* Syntax List

Table 71. i* Syntax Rules List

Rule	Description	i* wiki URL	Error Type	Supported by		
				Formalism	**Forward Eval**	**Backward Eval**
Actor within Actor	Reports when an actor, agent, or role is within another actor, agent, or role	http://istar.rwth-aachen.de/tiki-index.php?page_ref_id=233	Error	Not specified	Not specified	Not specified
Anon association and dependency link used between Actors	Reports when a link that is not an association or dependency link is used between two Actors		Error	No	No	No
Dependency link inside an actor	Reports when a Dependency link is found within an Actor	http://istar.rwth-aachen.de/tiki-index.php?page_ref_id=238	Error	Not specified	Not specified	Not specified
Dependency link connected to Actor in SR diagram	Reports when a Dependency link is connected to an expanded Actor in an SR diagram	http://istar.rwth-aachen.de/tiki-index.php?page_ref_id=267	Warning	Not specified	Yes	Yes
Decomposition, Correlation, Means-Ends, or Contribution link is used between	Reports when a Decomposition, Correlation, Means-Ends, or Contribution is	http://istar.rwth-aachen.de/tiki-index.php?page_ref_id=274, http://istar.rwth-	Warning	Yes	Yes	Yes

Actor boundaries	extending over an Actor's boundary to another Actor.	aachen.de/tiki-index.php?page_ref_id=288				
Dependency link at actor boundary	Reports dependency links in an SR model that are connected with an actor instead of an element within an actor	http://istar.rwth-aachen.de/tiki-index.php?page_ref_id=267	Error	No	No	No
Specific actor links	Reports an actor/agent/role that should be modelled by a more specific actor symbol	http://istar.rwth-aachen.de/tiki-index.php?page_ref_id=232	Warning	Not Specified	Not Specified	Not Specified
Association links used between incorrect specialized Actors	Reports when an Association link is being used between to incorrectly specialized Actors		Warning	Not Specified	Not Specified	Not Specified
Goal or Softgoal is a leaf	Reports a goal or softgoal that is not decomposed any further	http://istar.rwth-aachen.de/tiki-index.php?page_ref_id=248	Warning	Yes	Yes	Yes
Softgoal not decomposed	Reports when a Softgoal is not being decomposed		Warning	Yes	Yes	Yes
Missing dependum	A dependency link does not show the dependum	http://istar.rwth-aachen.de/tiki-index.php?page_ref_id=241	Warning	Yes	Yes	Yes
Dependum used in more than one dependency	Reports a dependum that is used in more than one dependency	http://istar.rwth-aachen.de/tiki-index.php?page_ref_id=	Warning	Yes	Yes	No

		240				
Softgoal dependency on a Goal or Task	Reports when a Softgoal is dependent on a Goal or Task	http://istar.rwth-aachen.de/tiki-index.php?page_ref_id=235	Warning	Yes	Yes	Yes
Both sides of Dependency Links point in different direction	Reports when a intention has two Dependency links and they point in different directions	http://istar.rwth-aachen.de/tiki-index.php?page_ref_id=239	Warning	Yes	Yes	Yes
Goal not decomposed by means-end link	Reports when a Goal is being decomposed by a link that is not a means-end lnk	http://istar.rwth-aachen.de/tiki-index.php?page_ref_id=210	Warning	Yes	Yes	Yes
Means-end used improperly	Reports when a Means-ends link is not used from a Task to a Goal	http://istar.rwth-aachen.de/tiki-index.php?page_ref_id=271, http://istar.rwth-aachen.de/tiki-index.php?page_ref_id=208	Warning	Yes	Yes	Yes
Correlation or Contribution used improperly	Reports when Correlation or Contribution is not used from any element to a Softgoal	http://istar.rwth-aachen.de/tiki-index.php?page_ref_id=289, http://istar.rwth-aachen.de/tiki-index.php?page_ref_id=285	Warning	Yes	Yes	Yes
Inconsistent Task Decomposition links	Reports when a Task's Decomposition links are inconsistently directed	http://istar.rwth-aachen.de/tiki-index.php?page_ref_id=	Warning	Yes	Yes	Yes

	between itself and a child	272, http://istar.rwth-aachen.de/tiki-index.php?page_ref_id=273				
Decomposition Link used incorrectly	Reports when a Decomposition Link is used from Goal to Tasks, Softgoal, or Resources, or from Softgoal to Task.		Warning	Yes	Yes	Yes
Means-Ends link used between Tasks.	Reports when a Means-Ends link is used between two Tasks		Warning	Yes	Yes	Yes

Appendix B i* Syntax Variations Survey Papers

Table 72: List of References Surveyed in Chapter 5

Liu, L., Yu, E., Mylopoulos, J.: Security and Privacy Requirements Analysis within a Social Setting, In IEEE Joint Int. Conf. on Requirements Engineering, RE'03, pp. 151-161. (2003)
Gans, G., Jarke, M., Lakemeyer, G., Schmitz, D.: Deliberation in a metadata-based modeling and simulation environment for inter-organizational networks, In Proc. of the 15th Int. Conf. on advanced information systems engineering, CAiSE'03, pp. 587-607. (2005)
Samavi, R., Yu, E., Topaloglou, T.: Strategic reasoning about business models: a conceptual modeling approach. In Journal of Information Systems and E-Business Management (2008)
Grau, G., Franch, X., Maiden, N. A. M.: A goal based round-trip method for system development. In Proc. of the 11th Int. Workshop on Requirements Engineering: Foundation For Software Quality, REFSQ, pp. 71-86. (2005)
Raadt, V. D., Gordijn, J., Yu, E.: Exploring Web Services from a Business Value Perspective. In IEEE Joint Int. Conf. on Requirements Engineering, RE'05, pp. 53-62. (2005)
Lespérance, y., Lapouchnian, A.: On Using i* for Modeling Autonomy, Reasoning, and Planning in Adaptive Systems, Presentation in istar'08 Workshop (2008)
Oliveira, A. P. A., Prado Leite, J. S. C., Cysneiros, L. M.: AGFL - Agent Goals from Lexicon Eliciting Multi-Agent Systems Intentionality, Presentation in istar'08 Workshop (2008)
Kolp M., Faulkner, S., The SKwyRL Approach: Social/Spiral Design based on i* and Tropos, Presentation in istar'05 Workshop (2005)
Maiden N.A.M., Kamdar N. & Bush D.: Analyzing I* System Models for Dependability Properties: The Uberlingen Accident. . In Proc. of the 12th Int. Workshop on Requirements Engineering: Foundation For Software Quality, REFSQ, (2006)
Elahi, G., Yu, E.: A Goal Oriented Approach for Modeling and Analyzing Security Trade-Offs. In Proc. of 26th Int. Conf. of Conceptual Modeling, ER'07, pp. 375-390. (2007)
Strohmaier, M., Yu, E., Horkoff, J., Aranda, J., Easterbrook, S. M.: Analyzing Knowledge Transfer Effectiveness -An Agent-Oriented Modeling Approach, In Proc. of 40th Hawaii International Conference on Systems Science, HICSS'07, pp. 188 (2007)
Yu, E., Strohmaier, M., Deng, X.: Exploring Intentional Modeling and Analysis for Enterprise Architecture. In Proc. of the EDOC 2006 Conf. Workshop on Trends in Enterprise Architecture Research, TEAR 2006, (2006)
Mazón, J., Trujillo, J., Serrano, M., Piattini, M.: Designing data warehouses: From business requirement analysis to multidimensional modeling, In Proc. of Int. Workshop on Requirements Engineering for Business Needs and IT Alignment (2005)
Arzdorf, t., Gans, g., Jarke, m., Lakemeyer, g., Schmitz, d., SNet: A Modeling and Simulation Environment for Inter-Organizational Networks, Presentation in istar'05 Workshop (2005)
M. Kolp, T.T. Do, S. Faulkner, Multi-Agent Architecture for E-Business Systems: An Organizational Perspective, In Proc. of 23rd Int. Conf. on Information Systems, (2002)

Appendix C Forward Evaluation Example

Trusted Computing Example. In this example, we walk through the procedure, referring to the propagation rules used in each step of the algorithm. We apply the evaluation to the Figure 24 example, asking the question: "If the PC User Obtains PC Products from the Data Pirate, how does this effect the PC Product Provider's ability to Sell PC Products for Profit?" In order to reduce the space of the example, we will use element numbers to reference each element in the model, as shown in Figure 113.

Initiation: Place initial labels reflecting the analysis question. Add all initial labels into the label queue.

Initial LQ = {<i5, S>, <i4, D>, <i14, S>, <i16, S>}

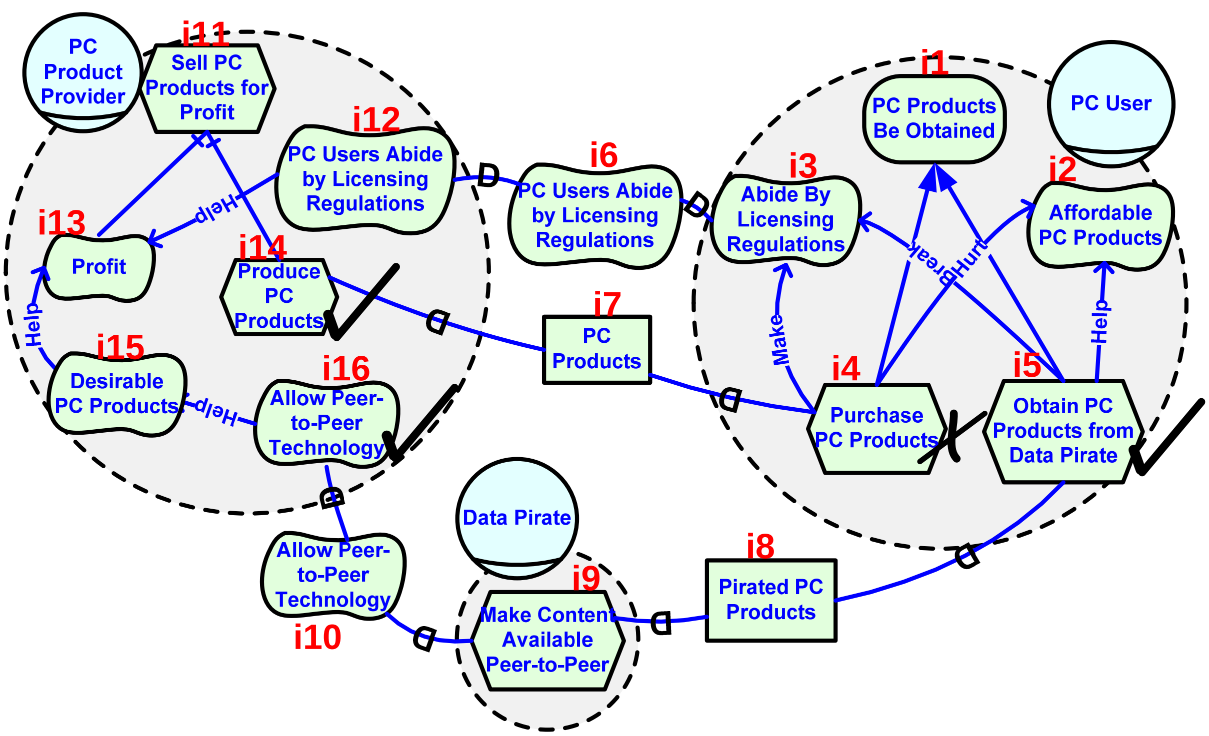


Figure 113 Simplified Trusted Computing (TC) Example with Numbered Intentions and Initial Evaluation Values

Iteration 1, Step 1:

Source Element, Label	**Rule**	**Result**
<i5, S>	Contribution Rules(S, Help)	i2.LabelBag = {<PS, i5>}
<i5, S>	Means-Ends Rule(S, D)	i1.v = S
<i5, S>	Contribution Rules(S, Break)	i3.LabelBag = {<D, i5>}
<i4, D>	Contribution Rules(D, Hurt)	i2.LabelBag = {<PS, i5>, <PS, i4>}
<i4, D>	Means-Ends Rule(S, D):	i1.v = S
<i4, D>	Contribution Rules(D, Make)	i3.LabelBag = {<D, i5>, <D, i4>}
<i14, S>	Dependency Rule(S):	i7.v = S
<i14, S>	Decomposition Rule(S, N)	i11.v = N
<i16, S>	Dependency Rule(S)	i10.LabelBag = {<S, i16>}
<i16, S>	Contribution Rules(S, Help)	i15s.LabelBag = {<PS, i16>}

LQ = {<i1, S>, <i7, S>}

Iteration 1, Step 2:

Softgoal	**Rule**	**Result**
i2	Automatic Cases(i2.LB, HJ)	N (No cases apply)
i2	Prompt User(i2.LB)	i2.v = S (User selects S) *
i3	Automatic Cases(i3.LB, HJ)	i3.v = D (Case 2)
i10	Automatic Cases(i10.LB, HJ)	i10.v = S (Case 1)
i15	Automatic Cases(i15.LB, HJ)	i15.v = PS (Case 1)

* In the case where the PC User Obtains PC Products from the Data Pirate and does not Purchase PC Products, the evaluator decides that PC Products are fully affordable.

LQ = {<i1, S>, <i7, S>, <i2, S>, <i3, D>, <i10, S>, <i15, PS>}

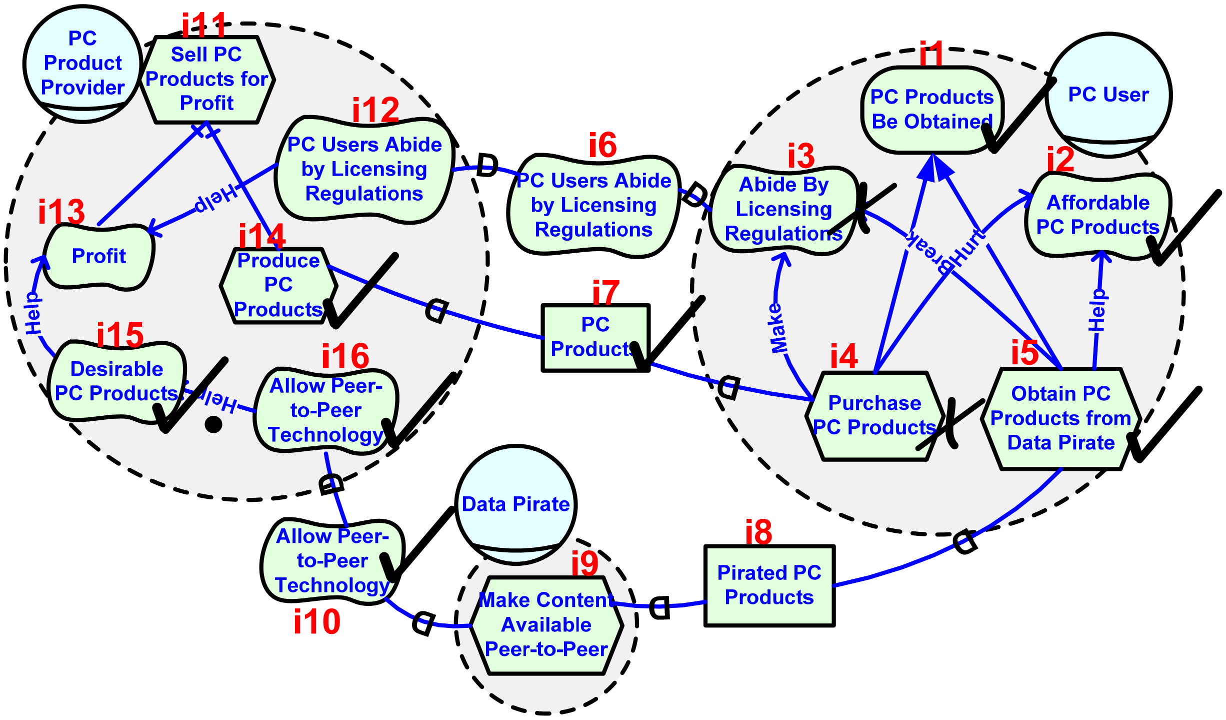


Figure 114 Simplified TC Example after Iteration 1

Iteration 2, Step 1:

Source Element, Label	Rule	Result
<i1, S>	N/A	None
<i7, S>	N/A	None
<i2, S>	N/A	None
<i3, D>	Dependency Rule(D)	i6.LabelBag = {<D, i3>}
<i10, S>	Dependency Rule(S)	i9.v = S
<i15, PS>	Contribution Rules(PS, Help)	i13.LabelBag = {<PS, i15>}

LQ = {<i9, S>}

Iteration 2, Step 2:

Softgoal	Rule	Result
i6	Automatic Cases(i6.LB, HJ)	i6.v = D (Case 1)
i13	Automatic Cases(i13.LB, HJ)	i13.v = PS (Case 1)

LQ = {<i9, S>, <i6, D>, <i13, PS>}

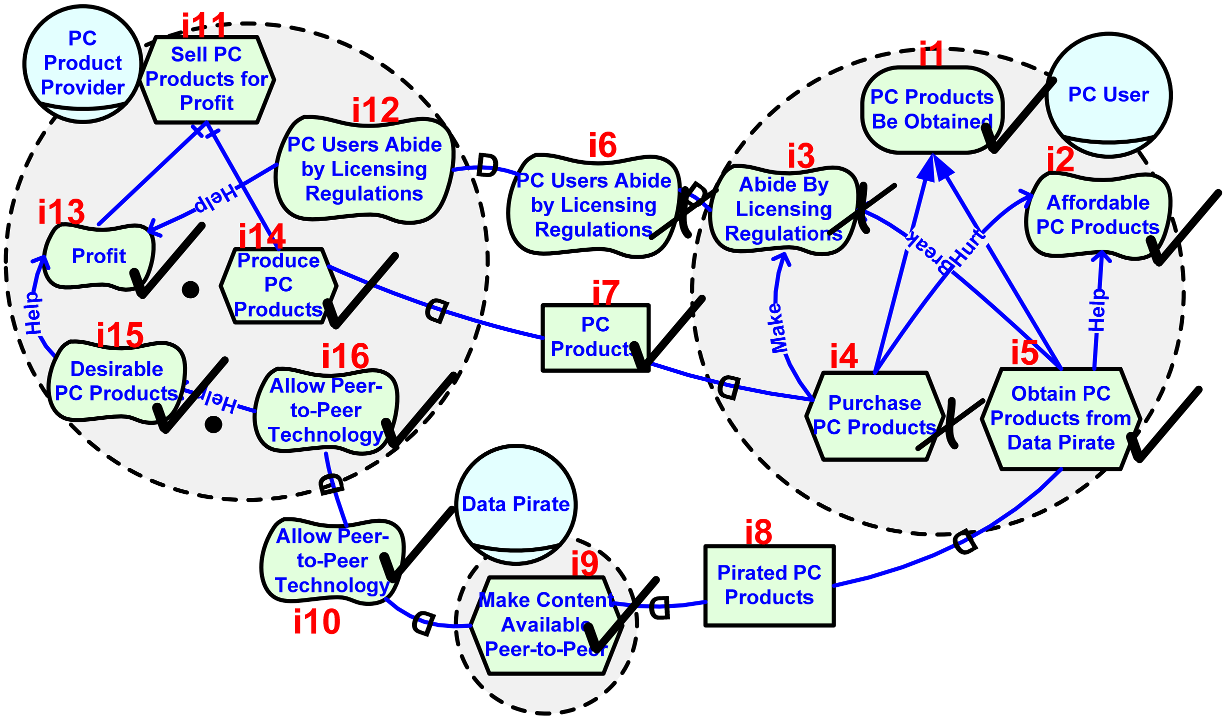


Figure 115 Simplified TC Example after Iteration 2

Iteration 3, Step 1:

Source Element, Label	**Rule**	**Result**
<i9, S>	Dependency Rule(S)	i8.v = S
<i6, D>	Dependency Rule(D)	i12.LabelBag = {<D, i6>}
<i13, PS>	Decomposition Rule(S, PS)	i11.v = PS

LQ = {<i8, S>, <i11, PS>}

Iteration 3, Step 2:

Source Element, Label	**Rule**	**Result**
i12	Automatic Cases(i12.LB, HJ)	i12.v = D (Case 1)

LQ = {<i8, S>, <i11, PS>, <i12, D>}

Iteration 4, Step 1:

Source Element, Label	**Rule**	**Result**
<i8, S>	N/A	None

<i11, PS>	N/A	None
<i12, D>	Contribution Rules(D, Help)	i13.LabelBag = {<PS, i15>, <PD, i12>}

LQ = {}

Iteration 4, Step 2:

Softgoal	**Rule**	**Result**
i13	Automatic Cases(i13.LB, HJ)	N (No cases apply)
i13	Prompt User(i13.LB)	i13.v = D (User selects D) *

* In the case where PC Users do not Abide by Licensing Regulations and Products are partially Desirable, the Desirability of products is not very effective in prompting PC Users to abide by regulations and Profit is denied.

LQ = {<i13, D>}

Iteration 5, Step 1:

Source Element, Label	**Rule**	**Result**
<i13, D>	Decomposition Rule(D, S)	i11.v = D

LQ = {<i11, D>}

Iteration 5, Step 2: Nothing Happens

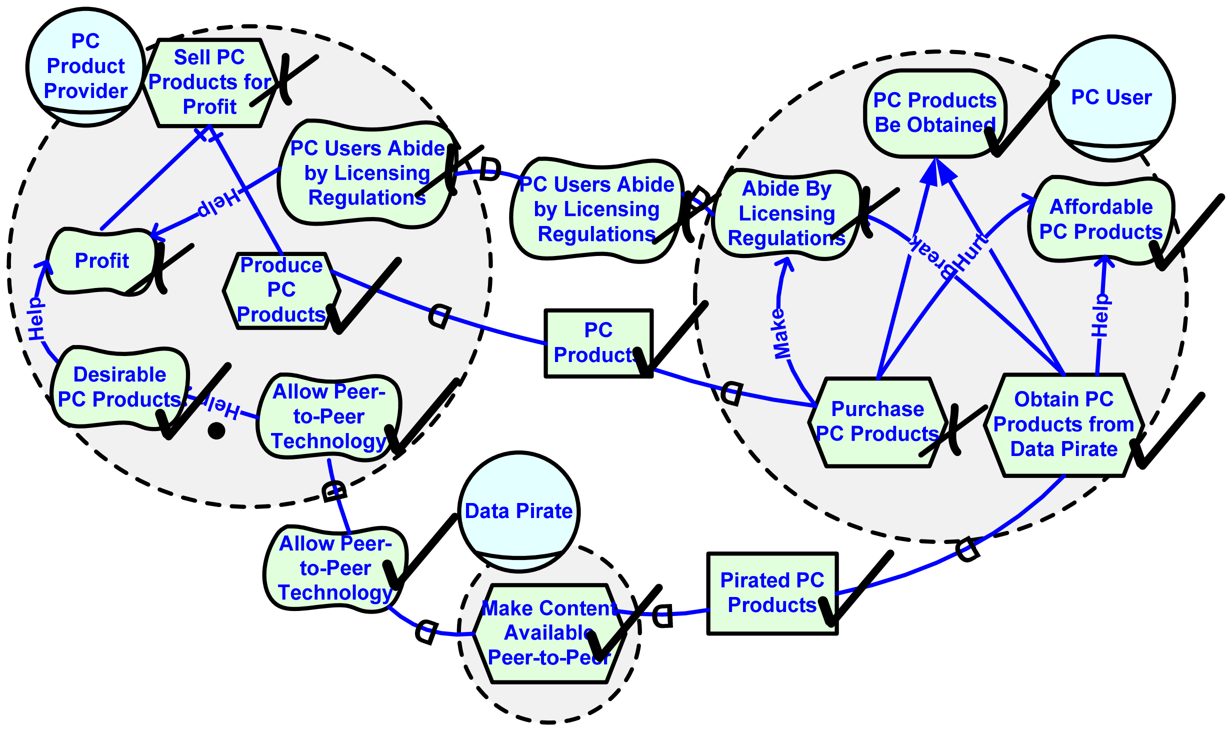


Figure 116 Simplified TC Example showing Final Evaluation Results

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 iteration on the model are needed. Ideally, a solution would be found where the PC Product Provider can make and Profit and the PC User can have Affordable products while Abiding by Licensing Regulations.

Appendix D   Additional Implementation Details

Table 73 Students who have Contributed to OpenOME under the Supervision of the Author

Name	Year	Project
Monica Olinescu	2008	General functionality, adding labels
Kelvin Ng	2008-09	General functionality, collapsing actors
Johan Harjono	2009	General functionality
Lee Yong Woo (Alfred)	2009	Viewing Judgments
Aftab Sultan	2009-10	Alternatives Tab, Saving analysis results
Fahad Fayyaz	2010	Human Judgment Changes
Arup Ghose	2010	Evaluation Usability
Michael Zammit	2010	General functionality, bug fixes
Alexandru Margarit	2010	General functionality, bug fixes
Aleli Evangelista	2011	Human judgment view, consistency checks, testing, bug fixes
Showzeb Ali	2011	Import/Export, testing, bug fixes
Osman Haque	2011	Syntax checking, testing, bug fixes
Denys Pavlov	2011	Tabular view, testing, bug fixes

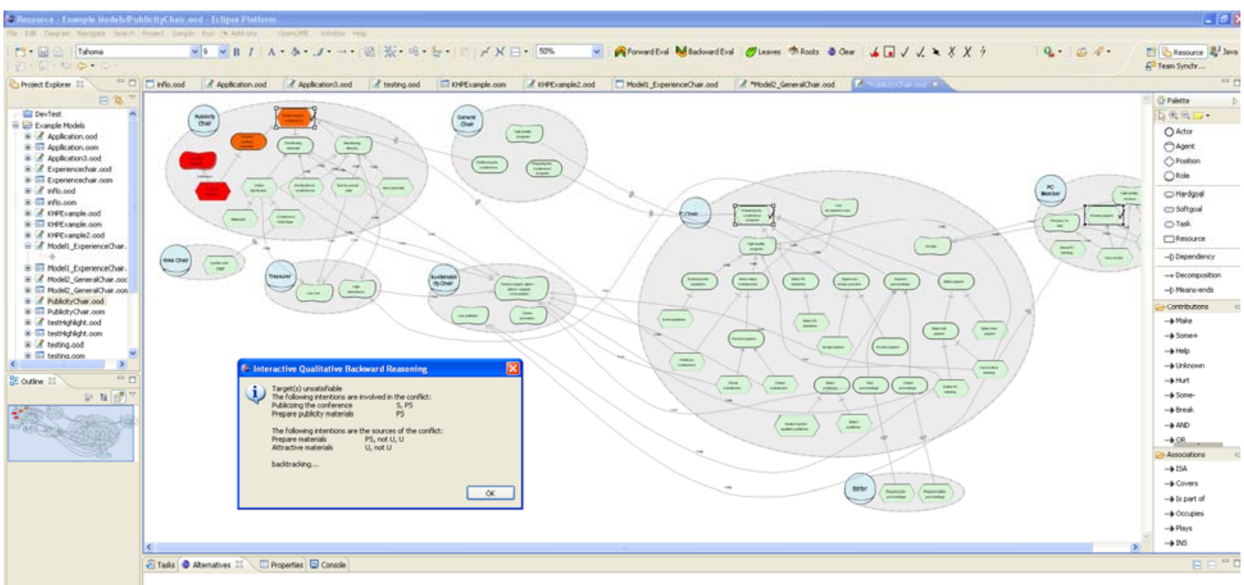


Figure 117 Example Conflict Highlighting in the OpenOME Tool

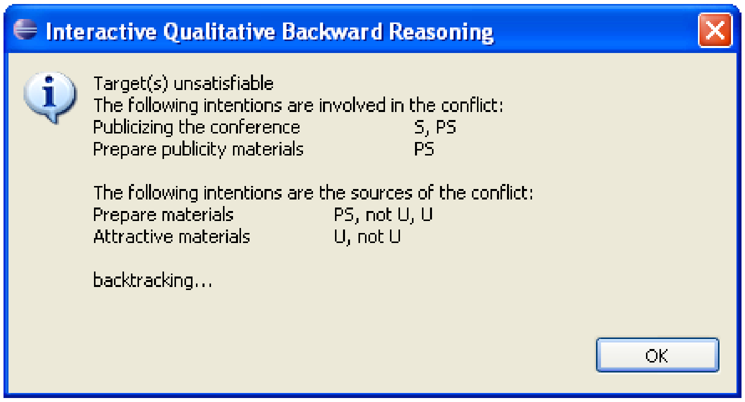


Figure 118 Example Conflict Pop-up in the OpenOME Tool

Appendix E Additional Validation Study Details

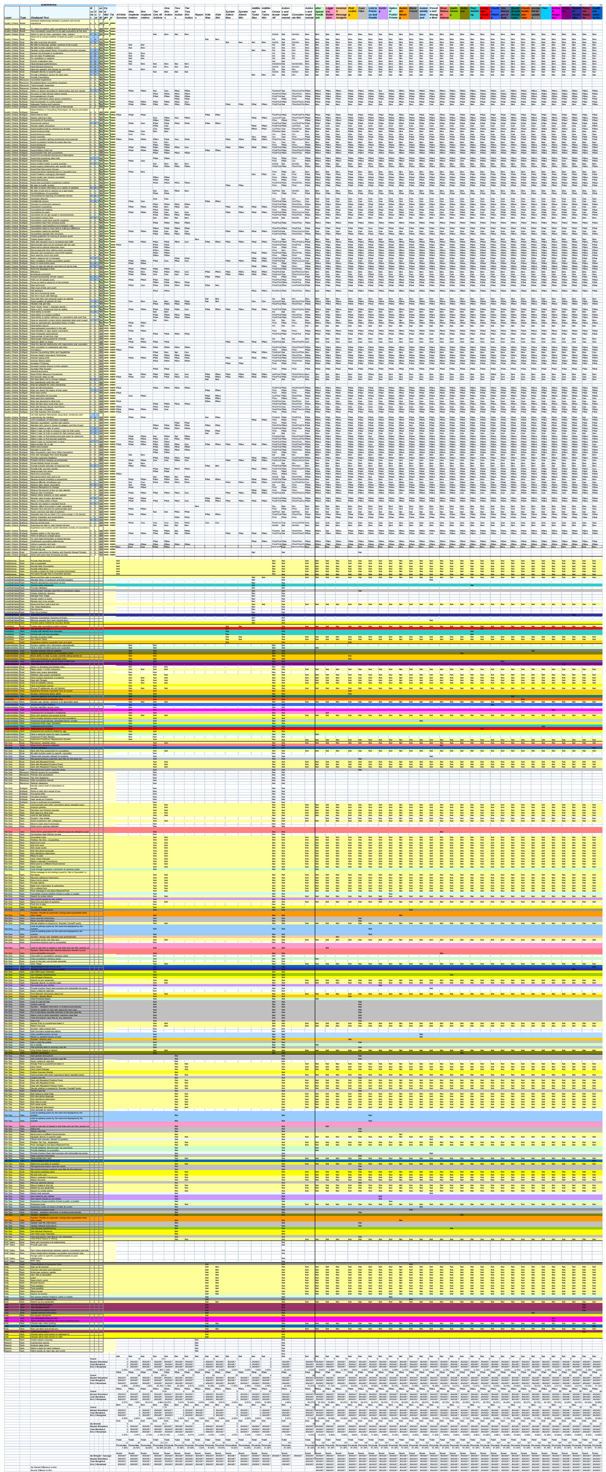


Figure 119 High-level View of a Table Showing the Forward Analysis Results over a large Model in the Counseling Service Case Study

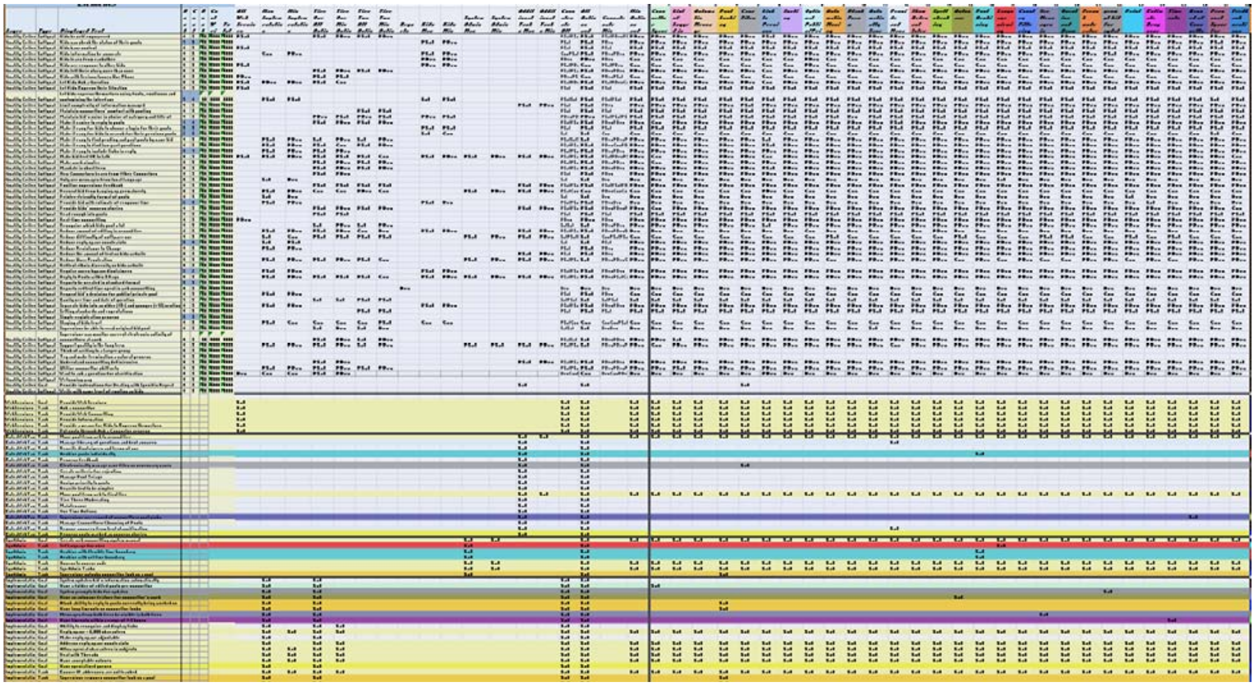


Figure 120 Closer View of a Table Showing the Forward Analysis Results over a large Model in the Counseling Service Case Study

Table 74 Individual Case Study Analysis Questions

Model 1	Forward	If every task of the Sustainability Chair and Local Chair is performed, will goals related to sustainability be sufficiently satisfied?
Model 1	Forward	If the Sustainability Chair does not perform any of their tasks, but both the Local and Conference Experience Chair perform all of their tasks, will goals related to attendee experience be sufficiently satisfied?
Model 1	Backward	What must be done in order to Encourage informal and spontaneous introductions and Make conference participation fun?
Model 1	Backward	In order to integrate newcomers to the ICSE community, what needs to be done? Are there things that the actors in the model can get away with not doing?
Model 2	Forward	If the cheapest/nicest hotel venue is chosen, will this have a significant effect on the success of the conference? What type of effect will it have?
Model 2	Forward	The General Chair depends on many actors. If all of the softgoals the General Chair depends on other actors for are somewhat satisfied, will the major goals of the General Chair be satisfied? Which ones are not satisfied, if any?
Model 2	Backward	Is it possible for successful conference to be fully satisfied? If so, how?
Model 2	Backward	Is it possible for both sustainability and successful conference to both be at least partially satisfied? If so, how?
Model 3	Forward	If the Publicity Chair distributes materials online and the PC Chair prepares only online proceedings and has only online submissions, how

		will this affect the significant goals of the actors (acceptance rate, quality of program, diffusion, etc.)?
Model 3	Forward	If the Publicity Chair distributes materials only at the conference and the PC Member uses an external reviewer, how will this affect the significant goals of the actors (acceptance rate, quality of program, diffusion, etc.)?
Model 3	Backward	Is it possible for both High Quality Program and Low Acceptance Rate to be satisfied, what choices in the model need to be made for this to happen?
Model 3	Backward	Is it possible to both Maximize Diffusion and have Low Cost? Can both these goals be at least partially satisfied? What else needs to be satisfied for this to happen?