Analyzing Goal Models – Different Approaches and How to Choose Among Them

Jennifer Horkoff
Department of Computer Science,
University of Toronto
40 St. George Street, Room 4242
Toronto, Ontario M5S 2E4
jenhork@cs.utoronto.ca

Eric Yu
Faculty of Information,
University of Toronto
140 St. George Street
Toronto, Ontario M5S 3G6
yu@ischool.utoronto.ca

ABSTRACT
A great variety of techniques for analyzing goal models in requirements engineering have been proposed in recent years. 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. From a practical viewpoint, this diversity creates a barrier for widespread adoption of such techniques. Recognizing the lack of guidance to the literature and how to choose among these techniques, this paper offers a first attempt to organize this body of knowledge and suggest initial guidelines on choice of techniques to meet users' analysis objectives.

Categories and Subject Descriptors
D.2.1 [**Requirements/Specifications**]: *elicitation methods, languages, methodologies, tools.*

Keywords
Goal-Oriented Requirements Engineering, Model Analysis.

1. INTRODUCTION
Goal-Oriented Requirements Engineering (GORE) has received much attention in RE research as a means of understanding the underlying motivations for system requirements, helping to ensure that the right system is built to address the right problems, e.g., GORE techniques [2][5][9][10][32][36]. Goal models are unique among models used to capture a system domain and requirements in that their structure naturally leads to an analysis of the achievement of objectives as well other important domain properties such as security or trust. As such, work has suggested that we can gain further value from goal models by applying systematic analysis. However, many different analysis techniques for goal models have been introduced, taking a variety of approaches. Some techniques propagate satisfaction values through links to and from goals in the model [1][9][26], others apply metrics over the structure of the model [1][11], apply planning techniques using tasks and goals in the model [7][4], run simulations over model tasks [16][34], and yet others perform checks over model contents [14][19].

SAC'11, March 21-25, 2011, TaiChung, Taiwan.

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. In this work, we address two objectives: (1) survey available approaches for goal model analysis, and (2) provide initial guidelines for procedure selection. More specifically, we aim to answer the following questions:

1) *Survey of methods: What 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?*
3) *Information: What domain information is needed in order to use the methods?*
4) *Analysis Benefits: What are some of the potential benefits of goal model analysis in the requirements process?*
5) *Fitness for purpose: Which available methods can be applied to achieve which kinds of usage objectives?*
6) *Selection: How can we use this information to advise on selection?*

The benefits and guiding questions produced in this work are based on the authors' experience and are intended to provoke useful discussion. The guidelines are applied to several examples from the literature.

This paper is organized as follows: Section 2 provides background on goal modeling, Section 3 provides a survey of GORE analysis techniques, Section 4 enumerates potential benefits of goal model analysis, including guidelines for procedure selection and guideline application examples, and Section 5 provides conclusions and future directions.

2. BACKGROUND: GOAL-ORIENTED REQUIREMENTS ENGINEERING
Several different approaches using the concept of goals as part of a Requirements Engineering technique have been introduced. Generally, GORE frameworks allow for the representation of one or more stakeholder needs (goals), which may be assigned to an agent (stakeholder or system), and which may have relationships to other goals, often describing how a goal can be achieved. Example goal modeling frameworks, techniques, or methodologies include KAOS, GBRAM, AGORA, NFR, i*, Tropos, and GRL, 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 [10]. The GBRAM technique guides the elicitation of goals from system activities and artifacts, classifying goals, and associating them with constraints and scenarios [2]. Goals in GBRAM are refined using questions and scenarios, and are represented in tabular form. The Annotated Goal-Oriented Requirements Analysis (AGORA) approach attempts to address 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.

The NFR (Non-Functional Requirement) modeling aims to represent human intentions in technical systems [9]. The framework uses *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, made use of notations in the NFR Framework, including softgoals, AND/OR decompositions, and contribution links. To this it added tasks, (hard) goals, resources, and dependencies between actors (agents) [36]. The i* Framework was incorporated into the Tropos Framework, using goal models as part of an agent-oriented system development methodology starting with goal models [5]. A reduced version of i* was used to create GRL (Goal-oriented Language), used with Use Case Maps (UCM) as part of URN (User Requirements Notation), now an ITU-T (telecommunications) standard [32].

In this work, we focus on systematic analysis procedures over primarily graphical goal model representations consisting of goals and relationships. To be considered in our survey, an analysis procedure must work over models which minimally support a set of goals linked together by AND/OR 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 a security. As a result of our focus on graphical AND/OR models, most of the procedures surveyed in the following sections work over models represented in one or more of the KAOS, NFR, i*, Tropos, or GRL Frameworks.

3. SURVEY OF GOAL MODEL ANALYSIS TECHNIQUES

We provide an overview of GORE analysis techniques. The focus is on techniques which analyze a model after its creation, as opposed to techniques which direct the creation of models. 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 were 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 finding shave 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 GORE analysis work.

The remainder of this Section provides an overview of GORE analysis approaches, grouping them in categories according to the techniques used. We use this categorization as it is closely related to the type of analysis questions facilitated by the procedures.

3.1 Satisfaction 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), [1][9][21][22][23][26][33], or backward, [22][23][27][31]. These procedures can answer questions like “What is the effect of this alternative? (forward)” or “Can these goals be satisfied? (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, [1][9][23][26][27]. Other procedures produce binary results, where goals have only one of two values, typically satisfied or not. For example, Maiden et al. analyzes in terms of compliance, whether an argument can be made to justify the satisfaction of tasks and resources based on existing requirements [33].

Several procedures offer quantitative analysis, using numbers to represent the probability of a goal being satisfied or denied [21][31], or to represent the degree of satisfaction/denial [1]. The backwards approach in [22] allows for the addition of analysis constraints, conflict restrictions, and finding a minimum cost solution. Asnar & Giorgini [3] expand on [23] to include quantitative analysis of acceptable risk levels and costs. This procedure works over an expansion of an existing goal model framework (Tropos) which includes events, risks, and (risk) treatments. Wang et al. [35] adapt the work of Giorgini et al. [22], using goal models to diagnose run-time failures. Amyot et al. [1] use 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.

One of the 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 [21][22][23]. Other procedures make use of predefined rules to combine multiple values [1]. Further procedures are “interactive”, using human intervention based on domain knowledge to resolve partial or conflicting evidence [9][26][27].

3.2 Metrics

Several approaches aim to measure qualities over the domain, such as security, vulnerability, and efficiency, using metrics over constructs in the model. 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?" In order to help in the selection of alternative components and architectures, Franch & Maiden [12] use counts of dependency classifications (for e.g., instance, model, duplicate, hidden) in an Strategic Dependency (SD) model (actors and dependencies only) as part of quantitative formulas aimed to calculate vulnerability, packaging, self-containment, uniformity, and connectivity. Franch et al. [13] continue this work by introducing the means to calculate global or local metrics over SD models using classifications and weights of actors and dependencies in an SD model. This work is expanded in [11] to work over both SD and Strategic Rationale (SR, actor, goal, and dependency graphs), developing a framework which allows for qualitative or quantitative, automated or interactive metric calculation. Kaiya et al. [30] apply a similar approach in the Annotated Goal-Oriented Requirements Analysis (AGORA) method, using quality metrics over AND/OR goal trees annotated with construct rationale and goal priorities.

The metrics approach introduced in [13] and [11] has been applied in [25] to evaluate the effectiveness of alternative architectures discovered via a systematic process, and in the PRiM approach [24] to find the best process alternatives.

3.3 Planning

Methods have applied AI-type planning to find satisfactory sequences of actions or design alternatives in goal models. These procedures can be used to answer questions such as "What actions must be taken to satisfy goals?" or "What are the best plan of actions according to certain criteria?" For example, Bryl et al. [6] 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 [11]. Plan discovery requires the definition of axioms that define possible goal decompositions and delegations, a definition of the capabilities of individual actors in the model. The procedure stops when an acceptable (but not necessarily optimal) plan is found. The method suggests that non-functional requirements may be considered via rules integrated into global criteria for plan selection.

This work is expanded in [7] as part of a systematic requirements analysis approach, including initial capability checks over the model and more general qualitative or quantitative criteria for evaluating plans. Such values can be resolved via designer expertise.

Asnar et al. [4] combine the planning approach of Bryl et al. [7] with the analysis of risk in Asnar & Giorgini [3], adding formalisms for measuring and relaxing criticality, the minimum level of trust required for delegation. Designer intervention is used to allow exceptions in the automatic plan refinement procedure. The approach uses the qualitative analysis from [22].

3.4 Simulation

Several approaches have added temporal information to goal models to allow for simulation over the network represented by model constructs. 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 alternatives is selected?" Gans et al [18] extend goal (i*) models represented in the Telos requirements language with temporal information including pre- and post-conditions to form the SNet Framework, converting parts of the models, excluding softgoals and contributions, into ConGolog programs (situation calculus). Once a model has been translated, the behavior of model actors can be simulated. During a simulation, a user can invoke exogenous actions interactively. Their use of a ConceptBase metadata manager based on Telos to represent extended i* models allows them to perform static checks on the model.

Wang and Lesperance [34] take a similar approach, but differ in the specifics of the mapping between i* and ConGolog. This approach introduced annotated i* SR diagrams (ASR), making use of composition and link annotations. Composition annotations consist of sequence, alternative, concurrency, and prioritized concurrency. Link annotations indicate conditions for the execution of the subtask, and the number of times it should be performed (cardinalities).

Gans et al. [18] extend results from [18] to incorporate a decision-theoretic planner into 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. The approach is further expanded in [16] to include roles, monitoring of delegations and evolution of agents. Roles are used to cover redundant capabilities of actor instances, parameterized by the duration of the tasks they perform and the contribution towards softgoals. Monitoring is performed using utility functions over softgoal contributions, with actions potentially taken after expectations are compared to real measures.

Gans et al. [17] expand the same framework to consider trust in individuals, confidence in a network, and distrust in both as quantitative measures. The measures can be viewed as quantitative metrics over the model. In addition to use of goal hierarchies mapped to ConGolog, the Action Workflow speech-act framework is used, describing the agent cooperation process in loops of communicative actions. Speech acts refining plans are interlinked to models of trust confidence and distrust, used together to make decisions for agents. Unlike [18], this work 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.

3.5 Model Checking

Several 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?" We have already summarized several approaches, such as [7] and [16], which combine the use of model checking with planning or simulation. However, work exists which is devoted entirely to checks over goal models. In [15] and [14], Fuxman et al. convert i* models to Formal Tropos, which includes formal expressions of creation, fulfillment and invariant properties. Temporal ordering (prior-to) and cardinalities are added to goal relationships. The translation of i* to formal Tropos is partially automated using conversion rules. The models are supplemented 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.

Giorgini et al. [20] extend i*/Tropos to better handle security and trust, separating trust dependencies from functional dependencies, distinguishing ownership and considering the delegation of permissions. They represent these ideas using formal predicates and check their models using Datalog, which accepts a logic program composed of a set of rules representing the model. Checks are performed for consistency, making sure there are no contradictions, after which trust and delegation in the model is checked for correctness.

Bryl et al. [8] combine the model checking approach in [19] with the planning approach in [6]. They consider privacy and security restrictions in the planning process and argue for the automatic derivation and selection of design alternatives early in the system development process, as a means to produce a secure system.

3.6 Survey Summary

We summarize our survey results over several points. In the above, we organized the survey results by their approach (Satisfaction Analysis, Metrics, Planning, Simulation and Model Checking). However, this division is not clear-cut, as many techniques employ more than one approach. The algorithm approach taken by each of the works mentioned is summarized in the first columns of Table 1. 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), No (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 procedure can produce different results in more than one of these forms. For example, the NFR procedure produces only qualitative satisfaction analysis results [9], while the metric procedure in [11] can produce either qualitative or quantitative data. 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 Table 1.

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.

Table 1. GORE Analysis Methods Survey Summary

	Approach							Analysis Results			Additional Notation Supported		
Paper	**Satisf Forwds**	**Satisf Backwds**	**Human Interv**	**Metrics**	**Plan-ning**	**Simu-lation**	**Model Check**	**Qual**	**Quant**	**Binary**	**Depend-encies**	**Soft-goals**	**Contribution Links**
Chung et al. [9]	Y	N	Y	N	N	N	N	Y	N	Y	N	Y	Y
Giorgini et al. [21]	Y	N	N	N	N	N	N	Y	Y	Y	N	M	Y
Giorgini et al.[22]	Y	Y	N	N	N	N	N	Y	N	Y	N	M	Y
Giorgini et al. [23]	Y	Y	N	N	N	N	N	Y	N	Y	M	Y	Y
Horkoff & Yu [26]	Y	N	Y	N	N	N	N	Y	N	Y	Y	Y	Y
Maiden et al. [33]	Y	N	Y	N	N	N	N	N	N	Y	Y	Y	Y
Amyot et al. [1]	Y	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y
Asnar & Giorgini [3]	Y	Y	N	N	N	N	N	Y	M	Y	N	M	Y
Letier & vLams. [31]	Y	Y	N	N	N	N	N	N	Y	Y	M	N	N
Horkoff & Yu [27]	Y	Y	Y	N	N	N	N	Y	N	Y	Y	Y	Y
Wang et al. [35]	Y	Y	N	N	N	N	N	N	N	Y	N	M	Y
Franch & Maiden [12]	N	N	N	Y	N	N	N	N	Y	N	Y	Y	N
Franch et al. [13]	N	N	N	Y	N	N	N	M	Y	N	Y	Y	N
Franch [11]	N	N	Y	Y	N	N	N	Y	Y	N	Y	Y	Y
Kaiya et al. [30]	N	N	N	Y	N	N	N	N	Y	Y	N	N	M
Bryl et al. [6]	N	N	N	Y	Y	N	N	N	Y	Y	Y	N	N
Bryl et al. [7]	N	N	Y	Y	Y	N	Y	M	Y	Y	Y	N	N
Asnar et al. [4]	Y	Y	Y	Y	Y	N	N	Y	N	Y	Y	M	Y
Gans et al [18]	N	N	Y	N	N	Y	Y	N	N	Y	Y	N	N
Wang & Lesper. [34]	N	N	N	N	N	Y	N	N	N	Y	N	N	N
Gans et al. [16] [18]	N	N	Y	N	Y	Y	Y	N	Y	Y	Y	Y	Y
Gans et al. [17]	N	N	N	Y	N	Y	M	N	Y	Y	Y	N	N
Fuxman et al. [14] [15]	N	N	M	N	N	N	Y	N	N	Y	Y	Y	N
Giorgini et al.[20]	N	N	N	N	N	N	Y	N	N	Y	Y	N	N
Bryl et al.[8]	N	N	N	N	Y	N	Y	N	N	Y	Y	N	N

Table 2. Information Required by Each Procedure

	Additional Information	Required by
1	Goal Cost	Satisfaction Analysis: [23][4][22][3], Planning: [6]
2	Risk	Satisfaction Analysis: [3], Planning: [4]
3	Textual Arguments	Satisfaction Analysis:[33], Metrics, Model Checking: [30]
4	Probabilistic Information	Satisfaction Analysis: [23] [31]
5	Events and Treatments	Satisfaction Analysis: [3]
6	Importance/Priority	Satisfaction Analysis: [1], Metrics: [13] [1], Simulation: [34]
7	Actor Capabilities	Planning: [6] [7] [4], Model Checking: [8]
8	(Pre/Post) Conditions/ Temporal Information	Simulation: [34] [18] [18] [16] [17], Model Checking: [15] [14]
9	Delegation/Ownership	Model Checking: [19] [8]
10	Trust	Planning: [4], Simulation: [17], Model Checking: [20][8]
11	Speech Acts	Simulation: [17]
12	Confidence and Distrust	Simulation: [17]
13	Preferences	Model Checking: [30]
14	Cardinalities	Simulation:[34], Model Checking: [14]

The type of syntax supported is significant in that is affects the types of analysis questions which can be answered using the model. We explore the benefits of goal model analysis in the next section. Support for such syntax is summarized in the last columns of Table 1.

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. We list the additional information required for each procedure in Table 2.

The distinction between additional syntax and such additional information can be blurry, for example in the goal models used for the simulations in [16], 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.

4. GOAL MODEL ANAYSIS OBJECTIVES AND SELECTION GUIDELINES

By examining the capabilities of GORE analysis techniques described in our survey, we produced a list of categories for potential benefits gained through method application, namely: domain understanding, communication, model improvement, scoping, requirement elicitation, requirements improvement, and design. A mapping can be produced between these benefits and the approaches in our survey. In order to better motivate the mapping, several guiding questions are included with each benefit category, reflecting the capabilities of goal model analysis procedure. 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. 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 3 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 3 can be downloaded from [37].

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.

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.

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 [9] has claimed 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 [28]. 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 3, including guideline questions.

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 3.

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 3. 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 ([1][26][27][33]) i* Metrics ([11][12][13]) Tropos Metrics, Planning, or Model Checking ([4][6][7][8][14][15][19]) SNET([16][17][18])
	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 ([3][21][22][23]) KAOS Satisfaction Analysis ([31]) GRL Quant. Analysis ([1]) i* Quant. Metrics ([11][12][13]) Tropos Planning ([4][6][7][8]) Tropos Modeling Checking ([8][14][15][19]) SNET([16][17][18][18]) i* Simulation([34]), or Model Checking: Tropos ([8][14][15][19]) SNET([16][18])
Communication	QC1. Do you need to communicate with stakeholders? Validate requirements in the model? Justify recommendations?	**Yes. Try:** Forward Satisfaction Approaches: NFR([9]) Tropos([3][21][22][23]) KAOS([31]) i*([26][33]) GRL([1])
Model Improvement	QM1. Are you confident in the accuracy, structure, and completeness of domain knowledge and models?	**No. Try:** Interactive Approaches: NFR([9]) i*([26][27][33]) Tropos([4][7]) SNET([16][18]) i* Metrics([11])
	QM2. Would you like to verify critical properties over the model?	**Yes. Try:** Model Checking: Tropos([8][14][15][19]) SNET([16][18])
Scoping	QS1. Do you need to determine system scope?	**Yes. Try:** Agent Approaches: i*/GRL Satisfaction Analysis ([1][26] [27][33]) i* Metrics ([11][12][13]) Tropos Metrics, Planning, or Model Checking ([4][6][7][8][14][15][19]) SNET ([16][18])
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([9]) i*([27][27][33]) Tropos([4][7]) SNET([16][18]) i* Metrics([11])
	QE2. Do you need to find detailed system requirements?	**Yes. Try:** Quantitative or Detailed Information: Tropos Probabalistic Satisfaction Analysis ([3][21][22][23]) KAOS Satisfaction Analysis ([31]) GRL Quant. Analysis ([1]) i* Quant. Metrics ([11][12][13]) Tropos Planning ([4][6][7][8]) Tropos Modeling Checking ([8][14][15][19]) SNET([16][17][18][18]) i* Simulation([34])
	QE3. Do you need to consider non-functional requirements difficult to quantify?	**Yes. Try:** Approaches supporting softgoals or contributions: NFR([9]) i* Satisfaction Analysis ([26][27][33]) Tropos Satisfaction Analysis ([3][21][22][23]) Tropos Model Checking([14][15]) GRL([1]) i* Metrics([11][12][13]) SNET([16][17][18])
	QE4. Do you need to capture domain assumptions?	**Yes. Try:** Approaches using Satisfaction Arguments: i* Satisfaction Arguments [33]
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([31]) i* Metrics([11][12][13]) AGORA([30]) Tropos Risk, Trust, and Security([3][4] [8][19]) SNET Trust([17])
	QR2. Do you need to find errors and inconsistencies in requirements?	**Yes. Try:** Model Checking: Tropos([8][14][15][19]) SNET([16][18])
Design	QD1. Are you aware of a sufficient number of high-level design alternatives?	**No. Try:** Agent, Planning, Forward and Backward Satisfaction Approaches: NFR([9]) i* Satisfaction Analysis ([26][27][33]) Tropos Planning([4][6][7][8]) KAOS([31]) GRL Forward Satisfaction Analysis([1]) SNET Planning([16][18])
	QD2. Are you aware of a sufficient number of detailed design alternatives?	**No. Try:** Quantitative Planning, Forward and Backward Satisfaction Approaches: KAOS Satisfaction Analysis ([31]) GRL Forward Satisfaction Analysis([1]) Tropos Planning([6][7]) SNET Planning([16][18])
	QD3. Do you need to evaluate and choose between high-level design alternatives?	**Yes. Try:** Satisfaction Analysis, Metrics and Agent Approaches: KAOS Satisfaction Analysis([31]) i* Forward Satisfaction([26][33]) GRL Satisfaction Analysis([1]) i* Metrics([11][12][13]) Tropos Risk([4])
	QD4. Do you need to evaluate and choose between detailed design alternatives?	**Yes. Try:** Quantitative or Detailed Information: Tropos Probabalistic Satisfaction Analysis ([3][21][22][23]) KAOS Satisfaction Analysis ([31]) GRL Quant. Analysis ([1]) i* Quant. Metrics ([11][12][13]) Tropos Planning ([4][6][7][8]) Tropos Modeling Checking ([8][14][15][19]) SNET([16][17][18][18]) i* Simulation([34])
	QD5. Do you need to find acceptable processes?	**Yes. Try:** Planning Approaches: Tropos Planning([4][6][7][8]) SNET Planning([16][18])
	QD6. Do you need to test run-time operation before implementation?	**Yes. Try:** Simulation Approaches: SNET([16][17][18]) i* Simulation([34])

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 in [33] explicitly asks users to capture domain assumptions associated with system requirements in textual arguments associated with model evaluation.

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.

Design. Once a set of requirements have 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.

4.1 Guideline Usage Examples

We apply our guidelines to two of the case studies appearing in our surveyed papers: the Wireless Service described in [10] and the Counseling Service described in [9].

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 ([26][33]), GRL Satisfaction Analysis ([1]), Tropos Risk Analysis ([4]), and/or i* Metrics ([11][12][13]). The satisfaction analysis and metric techniques could be repeated or adjusted to specifically support privacy analysis.

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 ([26][33]). In further steps, models could be analyzed for anonymity or privacy with the same techniques or with GRL Satisfaction Analysis ([1]), and/or i* Metrics([11][12][13]). If the required detailed information is available, Tropos planning techniques could be used to find plans ([4][7][8], while other approaches could be used to simulate a process, SNET([16][17][18]) or i* Simulation([35]).

5. CONCLUSIONS

This paper is a first attempt to make goal model analysis techniques more accessible to modelers confronted with the great variety of techniques available. Our survey was not intended to determine which techniques are superior, but to enable the potential user to understand the unique abilities of each procedure, and to select appropriate analysis approaches. Although we are among the authors of several of the procedures under consideration we have attempted to be neutral in our analysis. A body of future work is needed to validate these guidelines in practice. We expect that the guidelines will be refined as more experiences from applications become available.

6. REFERENCES

[1] Amyot, D., Ghanavati, S., Horkoff, J., Mussbacher, G., Peyton, L., and Yu, E. 2010. Evaluating goal models within the goal-oriented requirement language. *International Journal Intelligent Systems* 25, 8 (Aug. 2010), 841-877.

[2] Antón, A. I., and Potts, C. 1998. The use of goals to surface requirements for evolving systems. In *Proceedings of the 20th international Conference on Software Engineering (ICSE'98)*. IEEE Computer Society, Washington, DC, 157-166.

[3] Asnar, Y. and Giorgini, P. 2006. Modeling Risk and Identifying Countermeasure in Organizations. In *Proceedings of the International Workshop on Critical Information Infrastructures*

Security (CRITIS'06). Springer, LNCS, 4347, 55-56.

[4] Asnar, Y., Bryl, V., and Giorgini, P. 2007. Using Risk to Evaluate Design Alternatives. In *Proceedings of Agent-Oriented Software Engineering (AOSE VII).* Springer, LNCS, 44405, 140-155.

[5] Bresciani, P. Perini, A. Giorgini, P. Giunchiglia, F., and Mylopoulos, J. 2004. Tropos: An Agent-Oriented Software Development Methodology. *Autonomous Agents and MAS*, 8, 3, 203-236.

[6] Bryl, V., Giorgini, P., and Mylopoulos, J. 2006. Designing Cooperative IS: Exploring and Evaluating Alternatives. In *Proceedings of Cooperative Information Systems (CoopIS'06).* Springer, LNCS, 4275, 533-550.

[7] Bryl, V., Giorgini, P., and Mylopoulos, J. 2007. Supporting Requirements Analysis in Tropos: a Planning-Based Approach. In *Proceeding of the Pacific Rim International Workshop on Multi-Agents (PRIMA'07)*. Springer, 5044, 243-254.

[8] Bryl, V., Massacci, F., Mylopoulos, J., and Zannone, N. 2006. Designing Security Requirements Models through Planning. In *Proceedings of the of the 18th International Conference on Advanced Information Systems Engineering (CAiSE'06)*. Springer, LNCS, 4001, 33-47.

[9] Chung, L., Nixon, B.A., Yu, E., and Mylopoulos, J. 2000. *Non-Functional Requirements in Software Engineering*. Kluwer Academic Publishers, Norwell, MA.

[10] Dardenne, A. Lamsweerde, A. van, and Fickas, S. 1993. Goal-Directed Requirements Acquisition. *Science of Computer Programming*, 20, 3-50.

[11] Franch, X. 2006. On the Quantitative Analysis of Agent-Oriented Models. In Proceedings of *Proceeding of the 18th International Conference on Advanced Information Systems Engineering (CAiSE'06)*. Springer, LNCS, 4001, 495-509.

[12] Franch, X. and Maiden. N.A.M. 2003. Modelling Component Dependencies to Inform Their Selection. In *Proceedings of COTS-Based Software Systems*. Springer. LNCS, 2580, 81- 91.

[13] Franch, X., Grau, G., and Quer, C. 2004. A framework for the definition of metrics for actor-dependency models. In *Proceedings of IEEE Joint Int. Conf. on Requirements Engineering (RE'04)*, IEEE Press. 327-328.

[14] Fuxman, A., Liu, L., Pistore, M., Roveri, M., and Mylopoulos, J. 2003. Specifying and analyzing early requirements: some experimental results. In *Proceedings International. Requirements Engineering Conference (RE'03)*. IEEE Press, 105- 114.

[15] Fuxman, A., Pistore, A., Mylopoulos, J., and Traverso, P. 2001. Model Checking Early Requirements Specifications in Tropos. In *Proceedings IEEE International Symposium on Requirements Engineering (RE'01)*. IEEE Press, 174-181.

[16] Gans, D., Schmitz, D., Arzdorf, T., Jarke, M., and Lakemeyer, G. 2004. SNet Reloaded: Roles, Monitoring and Agent Evolution. In *Proceedings International Workshop on Agent-Oriented Information Systems (AOIS'04).* Springer, LNCS, 3508, 68-84.

[17] Gans, G., Jarke, M., Kethers, S., and Lakemeyer, G. 2003. Continuous requirements management for organization networks: a (dis)trust-based approach. *Requirements Engineering*, 8, 4- 22.

[18] Gans, G., Jarke, M., Lakemeyer, G., and Schmitz, D. 2003. Deliberation in a Modeling and Simulation Environment for Inter-organizational Networks. In *Proceedings of the of the 15th International Conference on Advanced Information Systems Engineering (CAiSE'03)*. LNCS, 2681, 242-257.

[19] Gans, G., Jarke, M., Lakemeyer, G., and Vits, T. 2002. SNet: A modeling and simulation environment for agent networks based on i* and ConGolog. In *Proceedings of the of the 14th International Conference* on Advanced Information Systems Engineering (CAiSE'02). Springer, LNCS, 2348, 328-343.

[20] Giorgini, P., Massaci, F., Mylopoulos, J., and Zannone, N. 2004. Requirements Engineering meets Trust Management: Model, Methodology, and Reasoning. In *Proceedings of the International Itrust Conference*. Springer, LNCS, 2995, 176-190.

[21] Giorgini, P., Mylopoulos, J., Nicchiarelli, E., and Sebastiani, R.2004. Formal Reasoning Techniques for Goal Models. *Journal of Data Semantics*, LNCS, 2800, 1-20.

[22] Giorgini, P., Mylopoulos, J., and Sebastiani, R. 2004. Simple and Minimum-Cost Satisfiability for Goal Models. In *Proceedings of the of the 16th International Conference on Advanced Information Systems Engineering (CAiSE'04).* Springer, LNCS, 3084, 20-3-5.

[23] Giorgini, P., Mylopoulous, J., and Sebastiani, R. 2005. Goal-Oriented Requirements Analysis and Reasoning in the Tropos Methodology. *Engineering Applications of AI*, 18(2), 159-171.

[24] Grau, G., Franch, X., Maiden, N.A.M. 2008. PRiM: an i*-based process reengineering method for information systems specification. *Information and Software Technology*, 50(1-2), 76-100.

[25] Grau, G., Franch, X. 2007. A Goal-Oriented Approach for the Generation and Evaluation of Alternative Architectures. In *Proceedings of European Conference on Software Architecture 2007*. Springer, LNCS, 4758, 139-155.

[26] Horkoff, J., and Yu, E. 2009. Evaluating Goal Achievement in Enterprise Modeling – An Interactive Procedure and Experiences. *In Proceedings of the 2nd IFIP WG 8.1 Working Conf. on the Practice of Enterprise Modeling (PoEM'09)*. Springer, LNBIP, 39, 145-160.

[27] Horkoff, J., and Yu, E. 2010. Finding Solutions in Goal Models: An Interactive Backward Reasoning Approach. In *Proceedings 29th International Conference on Conceptual Modeling (ER'10).* Springer, LNCS, 6412, 59-75.

[28] Horkoff, J., Yu, E., and Ghose, A. 2010. Interactive Goal Model Analysis Applied - Systematic Procedures versus Ad hoc Analysis. In *Proceedings of 3rd IFIP Working Conference on the Practice of Enterprise Modeling (PoEM'10)*. In press.

[29] Horkoff, J., Yu, E., Liu, L. 2006. Analyzing Trust in Technology Strategies. In *Proceedings of International Conference on Privacy, Security and Trust (PST 2006)*. 21-32.

[30] Kaiya, H., Horai, H., and Saeki, M. 2002. AGORA: Attributed Goal-Oriented Requirements Analysis Method. In *Proceedings International Requirements Engineering Conference (RE'02)*, IEEE Press, 13.

[31] Letier, E. and Lamsweerde, A. van. 2004. Reasoning about Partial Goal Satisfaction for Requirements and Design Engineering. In *Proceedings of the International Symposium on the Foundations of SE (FSE'04)*. ACM Press, 53-62.

[32] Liu, L., and Yu, E. 2004. Designing information systems in social context: a goal and scenario modelling approach. In *Proceeding of the 14th International Conference on Advanced Information Systems Engineering (CAiSE'02)* (April 2004). Springer, 187-203.

[33] Maiden, N.A.M., Lockerbie, J. Randall, D. Jones, S., and Bush, D. 2007. Using Satisfaction Arguments to Enhance i* Modelling of an Air Traffic Management System. In *Proceedings of IEEE International Conference on Requirements Engineering (RE'07)*. IEEE Press, 49-52.

[34] Wang, X. and Lesperance, Y. 2001. Agent-oriented requirements engineering using ConGolog and i*. In *Proceedings of the Workshop on Agent-Oriented Information Systems*. 59-78.

[35] Wang, Y., McIlraith, S., Yu, Y., and Mylopoulos, J. 2007. An Automated Approach to Monitoring and Diagnosing Requirements. In *Proceedings International Conference on Automated Software Engineering (ASE'07).* ACM Press, 293-302.

[36] Yu, E. 1997. Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering. In *Proceedings of the 3rd IEEE International Symposium. on Requirements Engineering (RE'97)*, IEEE Press, 226-235.

[37] 2010. GORE Analysis Procedure Selection Table. www.cs.utoronto.ca/~jenhork/GOREAnalysisSelectionTable.zip