Evaluating Modeling Languages: An Example from the Requirements Domain

Jennifer Horkoff, Fatma Başak Aydemir, Feng-Lin Li, Tong Li, John Mylopoulos

University of Trento, Italy
{horkoff, aydemir, fenglin.li, tong.li, jm}@disi.unitn.it

Abstract. Modeling languages have been evaluated through empirical studies, comparisons of language grammars, and ontological analyses. In this paper we take the first approach, evaluating the expressiveness and effectiveness of *Techne*, a requirements modeling language, by applying it to three requirements problems from the literature. We use our experiences to propose a number of language improvements for *Techne*, addressing challenges discovered during the studies. This work presents an example evaluation of modeling language expressiveness and effectiveness through realistic case studies.

Keywords: Requirements Modeling, Empirical Evaluation, Goal Modeling, Model Reasoning, Trade-off Analysis

1 Introduction

Once a conceptual modeling language has been proposed, it must be evaluated. There are several approaches for doing so in the literature. For instance, [10] conducts a comparative evaluation of languages focusing on their underlying grammars, while, in the requirements domain, [20] evaluates language quality through empirical studies, and [12] adopts an ontological perspective by comparing the primitive concepts of the language to those of foundational or domain ontologies. Along a different path, language utility has been evaluated via experiments, often using student subjects, focusing on language comprehension (e.g., [22]) and/or the users' ability to carry out meaningful tasks (e.g., [15]). Other evaluations have focused on the effectiveness of a language's graphical syntax by comparison to standard principles (e.g., [21]), or through applications of the language to realistic examples or case studies (e.g, [9]).

In this study, we adopt the last type of language evaluation, studying the expressiveness and effectiveness of a language in capturing phenomena in three realistic case studies. Although existing work has advocated for an evaluation of language expressiveness (e.g., [10, 12]), evaluation has focused on theoretical expressiveness via comparison to grammars or ontologies. Here we focus on evaluating both expressiveness and effectiveness through realistic examples. Expressiveness measures the degree to which a language allows its users to capture phenomena in the domain. For our purposes, effectiveness measures the degree to which a language supports typical modeling tasks, in our case domain conceptualization and model reasoning for decision making. Although similar, our notion of effectiveness is broader than typical definitions of usability, including the ease of eliciting domain information needed to use language constructs. Our study constitutes an empirical evaluation – we argue that this type of evaluation is critical for evaluating a modeling language. As such, we provide an example of how to conduct such a study.

The subject of our study is the *Techne* Requirements Modeling Language (RML), an RML intended for modeling, stakeholder communication, and reasoning, first proposed in 2010 [16]. Since the 1990s, RMLs have modeled stakeholder requirements as goals, supporting an analysis that compares alternative solutions and trade-offs, especially among non-functional requirements captured as softgoals [4]. Existing analysis procedures for goal-oriented models, e.g. [4,15], allow analysts to evaluate alternatives, discover conflicts and determine the viability of particular alternative solutions. Despite the impact of such techniques, it has been argued that languages and approaches for qualitative goal model reasoning have several limitations [16], including a limited vocabulary of concepts and relationships (goal, softgoal, goal decomposition, etc.) and limited analysis power, often founded on coarse-grained qualitative reasoning. In their RE'10 paper, Jureta et al. [16] introduce the *Techne* goal-oriented RML, with the stated intention of addressing some of these limitations. The proposal offers an abstract RML, enriching goal model primitives with concepts from the core requirements ontology [17], including mandatory and preferred (nice-to-have) goals, domain assumptions, quality constraints, and priorities.

Although the concepts and reasoning techniques proposed by *Techne* have theoretical advantages over existing RMLs for expressing and analyzing a requirements problem, these advantages have not been fleshed out through case studies and applications of the *Techne* proposal. In contrast, goal-oriented languages such as *i** [23] have benefited from a body of evaluation work using a variety of evaluation strategies (e.g., [9,12,20–22]). Example *Techne* or *Techne*-style models have been provided as part of successive work on *Techne* [7,11]; however, these examples were created to illustrate particular uses of or extensions to the language, rather than for evaluation purposes.

In this paper we provide such an evaluation, testing the expressiveness and effectiveness of concepts and relationships (the grammar) provided by *Techne*, evaluating their ability to capture requirements phenomena in real domains. We focus on two particular tasks: model creation and reasoning in support of alternative selection trade-off analysis.

We measure expressiveness (*Ex*) of the *Techne* grammar by noting: how well the language grammar covers domain phenomena. In this study, we test the expressiveness (*Ex*) of *Techne* by going through three requirements analysis cases, noting when the language is not able to capture information or concepts deemed important for requirements analysis. We measure language effectiveness (*Ef*) by making note of: (*Efa*) how easy is it to use the grammar to capture domain phenomena, and (*Efb*) how easy is it to elicit and find information from domain sources corresponding to language concepts. In this work, we test the effectiveness of *Techne* by noting how easy or difficult it is to use the language to model the three selected requirements case studies, and to what degree the information needed for the models is readily available from the example sources. We further evaluate the effectiveness of *Techne* reasoning by determining to what degree it supports the selection amongst alternatives arising in the three case studies, particularly as compared to *i**-style analysis [15].

Each of the three studies applying *Techne* to realistic requirements analysis examples brought to light challenges with the *Techne* grammar. As such, before applying *Techne* again, we propose solutions to some of these challenges, and use these in subsequent studies. Thus, the paper presents a series of applications, challenges, and proposed solutions.

The contributions of this work include: (1) Evaluation of the expressiveness and effectiveness of *Techne* concepts and relationships; (2) Proposal of patterns that aid the transformation of *i** contribution links to *Techne* concepts; (3) Proposal of a systematic process for requirements elicitation by pinpointing information needed to facilitate *Techne* modeling and analysis; (4) An example study showing how to assess practical language expressiveness and effectiveness.

The rest of the paper is organized as follows. Sec. 2 provides background on goal modeling and *Techne*.Sec. 3 presents an overview of our three studies, while details, challenges, and solutions for the three studies are found in Sections 4, 5, and 6, respectively. We conduct a comparison of *i** and *Techne* analysis in Sec. 7. Related work is described in Sec. 8, while conclusions, threats to validity, and future work are discussed in Sec. 9.

2 Background

Goal Modeling. Goal modeling frameworks, such as the *NFR* (Non-Functional Requirements) and *i** (distributed intentionality) Frameworks [4, 23], include concepts such as *goals*, *softgoals* (objectives without clear-cut criteria for fulfillment), and *tasks*. Relationships include *dependencies* between actors, AND and OR (means-ends) *decomposition* of (soft)goals and tasks, and *contribution* links (*Make*(++), *Help*(+), *Hurt*(-), *Break*(–)) to capture qualitative trade-offs between non-functional requirements captured as *softgoals*. For example, Fig. 1 captures an example trade-off in the eTourism domain, where a Hotel can either rent or build a Computer Reservation System (CRS) in house, making a trade-off between Maximize profit and Facilitate control.

Several qualitative reasoning techniques have been introduced for goal models, e.g., [4, 15]. We can apply such techniques to our example model, evaluating alternatives, e.g., renting the CRS or developing it in house. Renting partially satisfies Maximize profit, but partially denies (provide negative evidence for) Facilitate control. The in house option produces the opposite effects. On the left side of the model, both the Quick-fix and Long-term web strategies alternatives have the same effect (*Help*, partial satisfaction) on Website usability/friendliness, as such it is impossible to choose between them. Our example illustrates a limitation of qualitative analysis. The model, as is, does not contain enough information to allow us to choose amongst alternatives. We must use our implicit domain knowledge to make decisions or enrich the model with more detail, i.e. further trade-offs to differentiate amongst alternatives.

Techne **Abstract RML.** The *Techne* RML, as introduced in [16], consists of several concepts: *goals* (g), *softgoals* (s), *tasks* (t), *domain assumptions* (k), and *quality constraints* (q). Together these concepts are called requirements. Requirements can be *mandatory* (M) or *preferred* (nice-to-have) (Pf). The framework provides three relations between elements: *inference* (I), *conflict* (C), and *priority* (P). If a premise element (g, s, q, or t), e.g., e_1, infers a conclusion element, e, this means that the achievement of e can be inferred from the achievement of e_1. Multiple premise elements, e.g., e_1 ... e_n, can infer the same conclusion element, e, and this is treated as a (non-exclusive) OR, where achievement of any e_1 ... e_n means e is achieved. Multiple

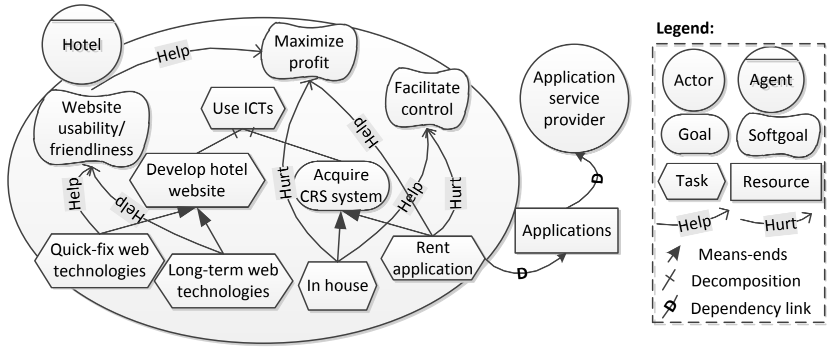

Fig. 1: Subset of an *i** model for eTourism

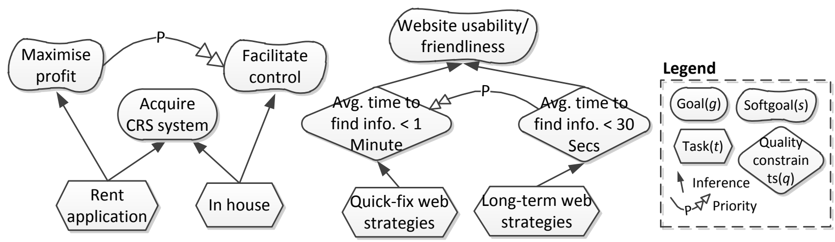

Fig. 2: Example *Techne* snippets addressing challenges shown in Fig. 1

premise elements can also be aggregated together to infer a conclusion element, e. The aggregation is described using functions of arbitrary complexity captured by associated domain assumptions (k). The most common aggregation is AND, meaning e_1 ... e_n must all be achieved for the e to be achieved. For simplicity, [16] suggests that a concrete syntax may be used to represent OR and aggregation via AND. As a shorthand we refer to OR and AND inferences throughout the paper.

Conflict (C) and priority (P) relations map a single element to a single element. Relating elements via conflict means that these elements cannot be satisfied simultaneously. Priorities between elements mean that one element has a higher priority than another. Note that we use preference and priority differently than in the original *Techne* proposal [16], which used optional and preference, respectively. Our terminology is more consistent with AI planning and later related work [19].

Reasoning with *Techne*. The *Techne* proposal outlines the discovery of candidate solutions, sets of tasks and domain assumptions which satisfy at least all mandatory requirements. Selection between these solutions can then be made using modeled priorities, although a detailed algorithm is not provided. Further work by Ernst et al. [8] suggests that solutions for a goal (*Techne*) model should first be ranked using priorities, and then should be ranked by the number of implemented preferred goals.

In theory, the richness of the *Techne* language should allow modelers to address the challenges illustrated with Fig. 1. In this case, a modeler could use priorities (Ps) between softgoals (Maximize profit and Facilitate control) to differentiate between options in the first alternative, and use quality constraints (qs) to distinguish the effects of each option in the second alternative (Fig. 2). [1] In this simple example, the optimal solution is now clear (Rent application and Long-term web strategies).

3 Case Study Setup and Overview

The studies were conducted in several interactive modeling sessions with three to four of the authors. All participants had experience in goal modeling, particularly with *i*/NFR* modeling. In each study, sources were used to collaboratively construct a goal model. Models for each study took a total of 6 - 12 hours to create. An overview of our process, including derived solutions and subsequent solution applications is shown in Fig. 3.

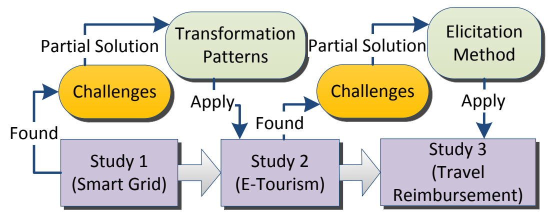

Fig. 3: Overview of Study Process

The first study evaluated the effectiveness of the *Techne* grammar via application to a Smart Grid example. This study revealed several challenges with *Techne* in practice (more details in Sec. 4). To address some of these challenges we proposed to start modeling using high-level concepts from *i*/NFR*, transforming these concepts to the richer ontology of *Techne* using developed patterns. The second study in the eTourism domain applied this proposal, testing the usability of our proposed patterns (Sec. 5). Here we uncovered challenges including difficulties in eliciting information needed for *Techne* modeling. We addressed this challenge by coming up with a method for systematically determining the information needed in order to choose between model alternatives. In the third study, focusing on a university travel reimbursement system, we applied this elicitation method, as well as the transformation patterns, recording our experiences (Sec. 6). Finally, we applied both *Techne* and *i** reasoning to our models, recording and comparing results (Sec. 7).

4 Study 1: Applying *Techne* to a Smart Grid Domain

The study focused on collaborative modeling of the Smart Grid domain, an information technology enhanced power grid, designed to optimize the transmission and distribution of electricity from suppliers to consumers [5].

Results. Statistics for the resulting model can be found in the first row of Table 1[2]. When modeling, we tried to capture the information provided in the source document without extensive extrapolation. As such, we drew no quality constraints (q) or domain assumptions (k), did not mark any goals as mandatory/preferred (M/Pf), and did not include any priority (P) links. In these cases, when information corresponding to a particular language concept was not readily available in our sources, a count of 0 was added to the table. The '-' symbol is used when a concept in the table was not part of an applied language (e.g., i* has no quality constraints (q), Techne has no decompositions (Dec)).

[1] In order to draw *Techne* models, we introduce a concrete visual syntax based on *i** syntax (see legend in Fig. 2 and Fig. 6).

[2] Full versions of all models available: www.cs.utoronto.ca/~jenhork/TechneEval/

Table 1: Model statistics for Study 1 to 3

	$Model$	a	g	s	t	r	k	q	I	C	P	M	Pf	Dec	Dep	ME	Hp	Ht	Mk	Bk	Un
Study 1	*Techne*	25	45	41	20	15	0	0	107	1	0	0	0	-	-	-	-	-	-	-	-
Study 2	*i**	31	22	65	52	7	19	-	-	-	-	-	-	52	23	12	35	9	22	7	1
	Techne	-	22	83	52	7	19	58	226	15	13	1	0	-	-	-	-	-	-	-	-
Study 3	*i**	6	42	8	30	1	0	-	-	-	-	-	-	36	7	30	5	4	0	0	0
	Techne	-	2	18	19	0	0	28	72	0	23	0	14	-	-	-	-	-	-	-	-

a: actor, r: resource, Dec: decomposition, Dep: dependency, ME: means-ends, Hp: help, Ht: hurt, Mk: make, Bk: break, Un: unknown

Challenges. *Availability of Information.* In this case, as shown in Table 1, our source document did not include information for identifying some *Techne*-specific concepts (q, k, P, M, Pf). This observation may change depending on available sources, or may be acquirable when interacting directly with stakeholders. We describe a proposed solution to this challenge in subsequent sections.

Representing Trade-offs. In drawing this initial *Techne* model, we had difficulty representing the notion of a trade-off between softgoals, finding the *Techne* concepts too absolute for modeling informal domains, which requirements problems usually are. In other words, it was difficult to capture weaker trade-offs between requirements using the binary, yes/no concepts offered by *Techne*. For example, the task Encrypt data had a positive effect on Communication Security but had a negative effect on Low cost technology. These effects could be best represented by the *Techne* inference (I) and conflict (C) relationships, respectively.

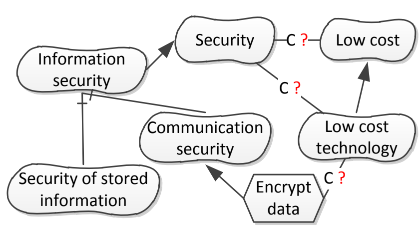

Fig. 4: Representing trade-offs in *Techne*

We found multiple possible placements for the conflict link representing the negative effect (see Fig. 4). However, unless Low cost is inferred by a further element, the three conflict possibilities have equal effect: in each case Low cost and Security cannot be simultaneously achieved when Encrypt data is implemented.

Using the *Techne* concept of priority, this model could have a solution. Either the users give higher priority to Low cost than they do to Security or vice versa, meaning that Encrypt data is not or is part of the solution, respectively. But asking users to express their priorities over such high-level goals may be problematic, most users would like to maximize both softgoals, instead of making a clear choice between them.

Solution: *i*-to-Techne* Transformation Patterns. This study has pointed out the following challenges with the effectiveness of the *Techne* grammar: (*Efa*) difficulty in expressing tradeoffs, particularly between softgoals, difficulty in selecting the most appropriate placement of conflict relations, and (*Efb*) several concepts are not easily determined on the basis of information available in our source. We address the *Efa* issue by developing a method which combines the expressiveness of *Techne* with the expressiveness of *i*/NFR*-style languages. The method starts with high-level *i*/NFR* style models with contribution links expressing trade-offs, and moves towards more precise *Techne*

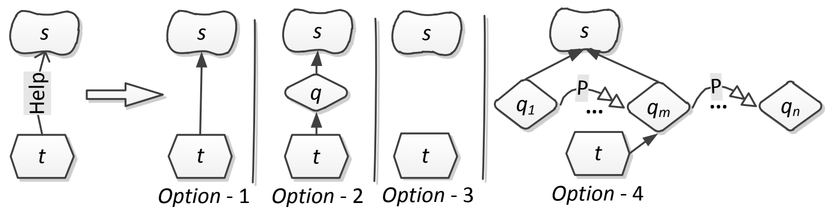

Fig. 5: *i**-to-*Techne* transformation pattern for *Help*

models, allowing for an eventual ranking of possible solutions. A solution to the *Efb* challenge is presented in Sec. 5.

We developed a series of patterns guiding users in transforming *i** contribution links to *Techne* concepts, mainly quality constraints and priorities. We show the transformation pattern for *Help* in Fig. 5. Patterns for other contribution links, *Make*, *Break* and *Hurt*, can be derived from this pattern, as follows. When transforming a *Make* contribution link, users can choose between the first two options: (1) replacing the link with an inference link, having the same semantics in both *i** and *Techne*, or (2) adding an inference link and a quality constraint, describing how the softgoal is approximated by a measure which is achieved by the contributing task/goal. The case for a *Break* link is similar, using a conflict link instead of an inference. As *Help*/*Hurt* represents the presence of partial positive/negative evidence, when transforming a *Help* or *Hurt* link, the user has two additional options (option 3 and 4): (3) ignore this evidence, indicating that it is not significant enough to retain in the model, or (4) capture this evidence as partial, meaning the contributing element infers a certain quality constraint (e.g., q_m, $m > 1$), but there are one or more quality constraints (e.g., q_1) with higher priorities which are not inferred by this element. In this case, *Hurt* may be distinguished from *Help* by the presence of quality constraints (e.g., q_n, $n > m$) with lower priorities. Although this process can be tool-supported, user judgment is required to select amongst transformation options and enhance the initial *i** model with additional required information.

Transformation from *i** to *Techne* may also involve refining softgoals into more detailed softgoals. If, for example, a softgoal has applicable quality constraints in multiple quality dimensions (time, cost, number of clicks, etc.) then the softgoal should be decomposed into more detailed softgoals expressing desires over each of these dimensions. We recommend that all quality constraints for a softgoal be in the same dimension. If this holds, then priorities between quality constraints can be derived automatically. E.g., if the softgoal is save time then the constraint q_1: time < 20 seconds is given higher priority than q_2: time < 1 minute automatically.

Other *Techne* specific information such as priorities not between quality constraints of the same dimension, domain assumptions or mandatory/preferred goals must be gathered using domain knowledge or further elicitation, i.e., this information cannot be derived from the *i** model.

5 Study 2: Applying Transformation Patterns to an eTourism Domain

Our second study focused on the eTourism domain, as described in [3]. We started by drawing a large *i** model covering the major sections of the document. Although we

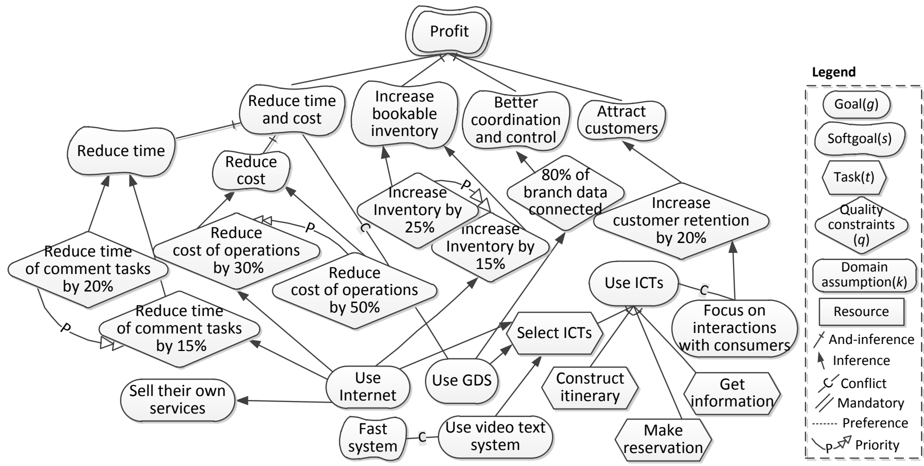

Fig. 6: Excerpt of the transformed *Techne* model for the Travel Agency actor in the eTourism domain

focused on *i** modeling, if we found any information corresponding to *Techne*-specific concepts (e.g., P, q, k), we included them in the *i** model, using our visual syntax. We then performed *i**-style qualitative analysis on this model, using the OpenOME tool [15], in order to evaluate the various alternatives in the model. Next, we applied our patterns in order to transform the model to *Techne*, creating models for each actor in the *i** model, noting challenges which arose. Finally, we conducted *Techne*–style analysis finding and ranking candidate solutions. Analysis results are discussed in Sec. 7.

Results. See Table 1 for statistics concerning the resulting models. In this case, we were able to come up with more *Techne*-specific concepts, including priorities, conflicts, and quality constraints, due to the application of our transformation patterns. We were also able to come up with several domain assumptions. We show a partial view of a model for the Travel Agency actor in Fig. 6.

Challenges. *Availability of Information.* When performing the transformation, we were generally able to come up with quality constraints for all softgoals; however, as this information was not available in our sources, our constraints were fairly arbitrary, making educated guesses concerning relevant measures and cut-off points. Priorities were also difficult to add using our own knowledge of the domain, unless they were over quality constraints in the same dimension. We made the same observations with designating goals as mandatory/preferred, although here we could assume the top-level goals of each actor, e.g. Profit in Fig. 6, were mandatory. Our experiences indicate that most of this information must be elicited directly from stakeholders.

Contributions between Softgoals. While our transformation patterns worked well for contributions between tasks and softgoals, they were difficult to apply to contributions between softgoals, occurring frequently in our *i** model. Such situations may result in inference links between softgoals or softgoals inferring quality constraints inferring softgoals. Although the latter is possible in *Techne*, it seemed peculiar to refine from soft to hard (quality constraint) to soft.

Contribution Aggregation. When combining together the effects of softgoals on a higher-level softgoal, even using our transformation patterns, we were left with choices between either AND or OR aggregated inference links, neither of which seem appropriate. For example, in Fig. 6, the top mandatory softgoal Profit is refined to several other softgoals including Increase bookable inventory and Reduce time and cost. Aggregating these softgoals together with OR was too weak, while using AND (as shown in Fig. 6) was too strong.

Techne provides a solution to this problem by allowing us to express an aggregation function of arbitrary complexity as part of the domain assumption associated with the aggregation of inference links. However, stakeholders must come up with a reasonable function that aggregates all inferring or contributing elements. How can Increase bookable inventory, Reduce time and cost and other softgoals be combined to determine the satisfaction of Profit? Such functions may be difficult for stakeholders to formulate.

Solution: Elicitation Method. The results of our second study helped us to identify challenges with the expressiveness and effectiveness of the *Techne* grammar: (*Ex*) it was difficult to capture contributions between softgoals, (*Efa*) it was difficult to capture and elicit complex aggregation of contributions, and (*Efb*) it was difficult to add accurate *Techne*-specific information to the models using information provided in our source. We propose a solution to address the *Efb* challenge, encountered in both the first and second studies, leaving the *Ex* and *Efa* challenges for future work. Specifically, we provide a list of suggested questions which can be asked to elicit *Techne*-specific elements:

1. Priority (P): For each pair of softgoals at the same level of decomposition, ask, "Which one do you prefer?" The answer results in a *priority* relationship.
2. Quality Constraint (q): For each softgoal, ask, "how can it be measured?" The answer results one or more quality constraints. If there is no answer, ask, "What do you mean by this softgoal?" The result is the refinement of the softgoal into more detailed softgoals.
3. Inference (I): For each new quality constraint or softgoal resulting from question 2, ask for each task, "Will this task satisfy the quality constraint or softgoal"? The answer introduces an inference link from the task to the quality constraint or softgoal.
4. Mandatory/Preferred (M/Pf): For each (soft)goal, ask if it is mandatory.

Asking such questions for all elements in a model would be laborious. Even for our simple Fig. 1 *i** model, we would have to ask 1, 3, at least 6, and 3 questions to elicit priorities, quality constraints, inferences, and mandatory/optional, respectively, a total of at least 13 questions. Thus we focus on eliciting only the information necessary to select between candidate solutions.[3] Starting with the high-level *i** model, we suggest the following method for targeted elicitation:

1. Apply *i** analysis (e.g., [15]) to evaluate possible alternatives (identified with OR inference links).
2. Use results of the analysis to determine if decisions can be made clearly for each OR alternative. Example cases where decisions cannot be made have been described in Sec. 2.
3. For each decision which cannot be made using *i** analysis, take a slice of the model starting with the OR alternative and moving up to all connected softgoals.

[3] When creating a complete requirements specification, further questions may be asked to elicit quality constraints for all softgoals, ensuring they are eventually measurable.

4. Convert each slice to *Techne*, combine slices together, if applicable. Conversion will apply transformation patterns as introduced in Sec. 4.
5. Apply questions as above to each element within this slice. The results of the patterns will determine whether which questions to ask stakeholders. For example, if *Option-2* or *4* in Fig. 5 are chosen, quality constraint questions should be asked.
6. Apply *Techne* reasoning to converted slices to determine the optimal model solution(s).

Step 5 may be further optimized by determining precisely which priorities and quality constraints are needed within a slice to choose between alternatives. We leave the specifics of such an algorithm to future work. We test our proposed solution via application to the third study.

6 Study 3: Targeted Elicitation in a University Travel Domain

Our third study captured the travel approval and reimbursement system of the University of Trento. For this case study, we interviewed four stakeholders, a Ph.D. student, a research project assistant, a secretary, and a research project leader who is also a professor. The first and last participants had knowledge of goal models. We first created an *i** model for three of the different interviewees, merged the model into a single model, then applied the elicitation method introduced in Sec. 5, returning to our stakeholders to ask follow-up questions targeting specific *Techne* constructs needed to support decision-making.

Results. See Table 1 for statistics describing resulting models. As the resulting *Techne* model focuses only on the slices needed to select among alternatives, the number of elements is relatively small. The initial *i** model had 116 elements and 102 links, while the slice resulting from our elicitation method had 70 elements and 95 links, reductions of 39.7% and 6.9%, respectively.

Our method resulted in a list of 52 questions for our stakeholders (10 P, 14 q, 14 I, and 14 M/Pf questions, respectively). Stakeholders were able to answer 7/10 of the P questions and all of the I and M/Pf questions. Of the 14 q questions asked, 2 questions elicited a softgoal decomposition, 1 question elicited a softgoal and a quality constraint, 8 questions elicited 16 quality constraints, while 3 questions could not be answered.

Challenges. Generally, stakeholders had some difficulty in quantifying softgoals into quality constraints, and were sometimes not able to provide priorities between softgoals. Often stakeholders responded with "it depends", meaning that there was some domain context not well-captured by the models. Our experiences emphasize the importance of perspective when drawing goal models, expressible using actors in *i**. The elicitation of quality constraints can differ greatly depending on the actor who desires the goal, e.g., Fast reimbursement time is evaluated very differently from the perspective of travelers and administrators.

The results of our third study helped us to identify challenges with the expressiveness and effectiveness of the *Techne* grammar: (*Ex*) the language did cover the actor concept well, important in capturing perspective; the language may require a richer means of capturing context; and (*Efb*) stakeholders sometimes had difficulty providing quality constraints and priorities.

7 Qualitative *i*/NFR* and *Techne* Reasoning Applied

As a point of comparison, we applied qualitative forward "what if?" *i** reasoning [15] to our *i** model in the second and third studies (we found no alternatives in the first study). That is, we placed initial qualitative values on model elements reflecting a particular decision over alternatives, then used *i** semantics to propagated these labels through the links, using human judgment to resolve conflicting or partial labels. The applied procedure is similar to other 'typical' qualitative goal model reasoning procedures, see [14] for a comparison.

In each model, we focused analysis on areas of the model connected to means-ends (OR) alternatives. We found three such decision-points in the eTourism Study and seven points in the University Travel Study. In some cases *i** analysis led to the clear selection of an alternative, while in other cases (see simple examples in Sec. 2) it was unable to clearly distinguish between available choices. We summarize these results in the third column of Table 2.

We also applied *Techne* reasoning as described in Sec. 2 to the models produced as a result of our transformations. In most cases, *Techne* analysis could find one or more clearly ranked solutions, making selections amongst alternatives. For example, in Fig. 2, a snippet from an eTourism study *Techne* model, there were a total of four possible solutions. The priorities added between quality constraints and softgoals allowed us to choose an optimal solution which includes both In house and Long-term web strategies. In other cases, solutions could not be ranked. For example, in Fig. 6, it was impossible to find a solution which satisfied the mandatory goal Profit, due to the AND-aggregated inference links between Profit and its refinements, and the inference and conflict links attached to Use GDS (i.e. Use GDS conflicts with Reduce time and cost while inferring Better coordination and control). *Techne* analysis results are listed in the fourth column of Table 2.

Table 2: Summary of *i** and *Techne* reasoning results

Decision-point	Actor	*i**	*Techne*
Quick-fix vs. Long-term	Hotel	CD	Long-term
Rent application vs. In house	Hotel	In house	In house
Internet vs. GDS vs. Videotext	Travel Agency	CD	Conflict: No solution
Ticket bought by agent vs. yourself	Student	CD	buy yourself
Hotel booked by agent vs. yourself	Student	CD	buy yourself
Book hotel vs. university acc.	Assistant	CD	Missing P: no solution
Cheap ticket vs. Ticket with short duration	Assistant	CD	short duration
Hotel close to the city center vs. conference center	Assistant	CD	close to conference center
Cheap ticket vs. Ticket with short duration	Professor	CD	short duration
Hotel close to city center vs. conference center	Professor	CD	close to conference center

CD: cannot distinguish

Our results show that the additional expressiveness provided by *Techne* allows the reasoning procedures to select amongst alternatives in more cases, when compared to the *i** procedure.

8 Related Work

As mentioned in the introduction, much work has been devoted to evaluating goal–oriented RMLs, notably the *i** Framework. The most relevant studies are those that focused on evaluation through case studies. For instance, Estrada et al. [9] conducted an empirical evaluation of the *i** modeling framework by using three industrial case studies. Their evaluation was based on a set of features that fall into two categories: modeling language (e.g modularity, expressiveness, and traceability) and pragmatics of the modeling method (e.g. scalability and domain applicability). The study reported good expressiveness and domain applicability, but poor modularity and scalability of *i**. In contrast, we have focused only on the expressiveness and effectiveness of the *Techne* grammar, including reasoning and decision-making, without focusing on scalability. Further case studies applying *i** in industrial contexts can be found as part of the iStar Showcase [1]. These studies typically focus on benefits and drawbacks of applying goal modeling in practice, without focusing on evaluating language expressiveness and effectiveness.

The *Techne* language has been applied and expanded in successive work, focusing, for example, on reasoning or product portfolio optimization [7, 11]. Some of these applications have introduced their own concrete syntax for *Techne*. Although their syntax was applied to illustrative examples in the paper, the focus was not on evaluating the usability of the syntax or the language, but on illustrating language expansion or reasoning for some further purpose. In [8], the authors provide intuitive guidelines for identifying preferred goals. For example, non-functional goals are ideal candidates for preferred goals, while top-level goals are typically mandatory goals. These guidelines can be useful to incorporate in our *i**-to-*Techne* patterns.

We have focused on assessing the reasoning powers of *Techne* compared to qualitative *i** reasoning. Other techniques for qualitative goal model reasoning exist, see [14] and [13] for relevant surveys. *Techne*'s use of priorities make it's reasoning capabilities unique when compared to other qualitative techniques, allowing for a final ordering of possible solutions. Yet other approaches have applied quantitative analysis procedures to such models, see [13] for examples. Although the analysis results produced by such procedures are more fine-grained, allowing users to better choose amongst alternatives, the accuracy of quantitative values propagated through goal models is suspect; furthermore, this information is difficult to acquire as part of early requirements analysis [6, 15].

Liaskos et al. [19] have proposed a concrete goal modeling language that uses preferences and priorities to choose between alternative solutions, bearing similarity to *Techne*. In their proposal, preference goals are prioritized using numerical weights obtained through *AHP* (Analytical Hierarchy Process). In this way, the challenge of eliciting numbers during early requirements is at least partially addressed. Accordingly, a solution that satisfies preference goals to a higher degree (i.e. the sum of the weights of satisfied preference goals) will be optimal. Further studies should evaluate whether this approach suffers from the challenges we have found for *Techne*, or whether numerical analysis with *AHP* is a feasible approach to find the aggregation function needed to evaluate high-level softgoals.

9 Discussion and Conclusions

We have provided an example study testing the expressiveness and effectiveness of the grammar of a conceptual modeling language using realistic cases. Specifically, we have conducted three studies evaluating the expressiveness and effectiveness of the *Techne* grammar. We have found several challenges relating to language expressiveness and effectiveness, including: (*Ex*) phenomena in the examples which are difficult to capture, including contributions between softgoals, actors, and context; (*Efa*) difficulty in expressing tradeoffs, placing conflict relations, and capturing complex aggregation of contributions; and (*Efb*) difficulty in finding or eliciting priorities, quality constraints, domain assumptions, and mandatory/optional goals. Overall, we find that *Techne*, as proposed, has reasonable expressiveness, but has significant challenges in regards to effectiveness.

We have addressed challenges related to language effectiveness by presenting two solutions: transformation patterns from *i** to *Techne*, allowing for easier capture of tradeoffs and prompting users to add *Techne*-specific constructs to their models; and elicitation methods focusing on targeted elicitation of *Techne*–specific constructs needed for decision-making.

This work has compared qualitative *i** to *Techne* reasoning, finding that *Techne* reasoning is able to choose between alternatives in more cases, providing enhanced reasoning power, given the availability of *Techne*–specific concepts.

Threats to Validity. *Internal.* In order to apply Techne, we have created a concrete syntax based on *i**. However, our purpose in this work was to evaluate the language grammar, not the visual syntax, thus we believe these choices did not impact our results. Our first two studies are limited by the nature of the sources, with information coming from a single document, without the opportunity to interact with stakeholders. Our third study is limited by its size, only interacting with four stakeholders, two of which were familiar with goal modeling. Challenges initially result from only a single study, although subsequent studies confirm discovered challenges in some cases.

External. We have selected domains representing complex socio-technical systems in order to evaluate the effectiveness of *Techne*. Although the choices of these domains may affect the ability to reproduce our results, we mitigate this threat via our use of three different domains. The studies were executed by researchers with goal model experience, hindering study repeatability. However, as this is (to our knowledge) the first study evaluating the effectiveness of *Techne* concepts, it is sensible to start with simpler domains and experienced users. The researchers applying the studies were also biased by their past experience using *i*/NFR*-style frameworks. However, as these frameworks have received much attention in research, it is reasonable to expect that many new *Techne* users would have a similar bias.

Future Work. We propose to look at techniques for expressing and eliciting complex aggregations for softgoals through an interactive, qualitative analysis in the spirit of [15]. We also plan to develop a tool–supported process that applies the transformation patterns proposed here into areas of the model critical for reasoning, automatically generating a list of questions for stakeholders covering missing information. Challenges in language expressiveness can be addressed by integrating *Techne* with existing goal–oriented approaches using actors [23] or context [2]. Combining the proposed *Techne* solutions with methods aimed at easing model elicitation, as in [18] and [6], may be another promising direction.

Acknowledgments. This work has been supported by the ERC advanced grant 267856, "Lucretius: Foundations for Software Evolution" (April 2011 - March 2016).

References

1. iStar Showcase'11: Exploring the goals of your systems and businesses, practical experiences with i* modeling, http://istar.rwth-aachen.de/tiki-index.php?page=iStar+Showcase%2711, 2011.
2. R. Ali, F. Dalpiaz, and P. Giorgini. A goal-based framework for contextual requirements modeling and analysis. *Requir. Eng.*, 15(4):439–458, Nov. 2010.
3. D. Buhalis and S. H. Jun. E-tourism. *Information Technology for Strategic Tourism Management*, 2003.
4. L. Chung, B. A. Nixon, E. Yu, and J. Mylopoulos. *Non-Functional Requirements in Software Engineering*, volume 5. Kluwer Academic Pub, 2000.
5. J. Cuellar and S. Suppan. A smart metering scenario, Network of Excellence on Engineering Secure Future Internet Software Services and Systems, eRISE 2013.
6. G. Elahi and E. Yu. Requirements trade-offs analysis in the absence of quantitative measures: a heuristic method. In *Proceedings of the 2011 ACM Symposium on Applied Computing*, pages 651–658. ACM, 2011.
7. N. Ernst, A. Borgida, J. Mylopoulos, and I. Jureta. Agile requirements evolution via paraconsistent reasoning. In *Advanced Information Systems Engineering*, pages 382–397. Springer, 2012.
8. N. Ernst, J. Mylopoulos, A. Borgida, and I. Jureta. Reasoning with optional and preferred requirements. In *ER 2010*, pages 118–131. Springer, 2010.
9. H. Estrada, A. Rebollar, O. Pastor, and J. Mylopoulos. An empirical evaluation of the i* framework in a model-based software generation environment. In *Advanced Information Systems Engineering*, LNCS, vol. 4001, pages 513–527. Springer, 2006.
10. A. Gemino and Y. Wand. A framework for empirical evaluation of conceptual modeling techniques. *Requirements Engineering*, 9(4):248–260, 2004.
11. J. Gillain, S. Faulkner, P. Heymans, I. Jureta, and M. Snoeck. Product portfolio scope optimization based on features and goals. SPLC, pages 161–170, 2012.
12. R. Guizzardi, X. Franch, G. Guizzardi, and R. Wieringa. Ontological distinctions between means-end and contribution links in the i* framework. In *Conceptual Modeling*, LNCS, vol. 8217, pages 463–470. Springer, 2013.
13. J. Horkoff and E. Yu. Analyzing goal models: Different approaches and how to choose among them. In *Proceedings of the 2011 ACM Symposium on Applied Computing*, SAC '11, pages 675–682. ACM, 2011.
14. J. Horkoff and E. Yu. Comparison and evaluation of goal-oriented satisfaction analysis techniques. *Requirements Engineering*, 18(3):199–222, 2013.
15. J. Horkoff, E. Yu, and A. Ghose. Interactive goal model analysis Applied-Systematic procedures versus ad hoc analysis. In *The Practice of Enterprise Modeling*, pages 130–144. Springer, 2010.
16. I. Jureta, A. Borgida, N. A. Ernst, and J. Mylopoulos. Techne: Towards a New Generation of Requirements Modeling Languages with Goals, Preferences, and Inconsistency Handling. *RE'10*, 2010.
17. I. Jureta, J. Mylopoulos, and S. Faulkner. Revisiting the Core Ontology and Problem in Requirements Engineering. In *RE'08*. IEEE, 2008.
18. S. Liaskos, R. Jalman, and J. Aranda. On eliciting contribution measures in goal models. In *RE'12*, pages 221–230. IEEE, 2012.
19. S. Liaskos, S. A. McIlraith, S. Sohrabi, and J. Mylopoulos. Representing and reasoning about preferences in requirements engineering. *Requirements Engineering*, pages 227–249, 2011.
20. R. Matulevičius and P. Heymans. Comparing goal modelling languages: An experiment. In *Requirements Engineering: Foundation for Software Quality*, pages 18–32. Springer, 2007.
21. D. L. Moody, P. Heymans, and R. Matulevicius. Improving the effectiveness of visual representations in requirements engineering: An evaluation of i* visual syntax. In *RE'09*, pages 171–180. IEEE, 2009.
22. M. A. Teruel, E. Navarro, V. López-Jaquero, F. Montero, J. Jaen, and P. González. Analyzing the understandability of requirements engineering languages for CSCW systems: A family of experiments. *Information and Software Technology*, 54(11):1215 – 1228, 2012.
23. E. Yu. Towards modelling and reasoning support for early-phase requirements engineering. In *RE'97*, pages 226–235. IEEE, 1997.