Scalable Modeling and Analysis of Requirements Preferences: A Qualitative Approach using CI-Nets

Zachary J. Oster
Department of Computer Science
University of Wisconsin-Whitewater
800 West Main Street
Whitewater, WI 53190
osterz@uww.edu

Ganesh Ram Santhanam
and Samik Basu
Formal Methods and Verification Group
Department of Computer Science
Iowa State University, Ames, IA 50011
gsanthan@iastate.edu, sbasu@iastate.edu

***Abstract*—We present a framework for reasoning with preferences in the context of Goal-Oriented Requirements Engineering (GORE). Our choice of preference language, conditional importance networks (CI-nets), is motivated by the occurrence in requirements engineering of qualitative preferences and tradeoffs involving sets of items; such preferences are expressed more naturally in CI-nets than in other representations. Building on our past experience with CI-nets, we are improving the scalability and usability of CI-nets for specifying and analyzing requirements preferences. We discuss our ongoing work and long-term plans, including efforts to develop more efficient methods to identify conflicting preferences among possible requirements, guide negotiation of resolutions to such conflicts, and improve traceability and comprehension of requirements preferences.**

I. Introduction

Identifying the requirements for a software system involves understanding and navigating tradeoffs among a wide variety of possible attributes that the system could possess. Many of these tradeoffs are *conditional*, meaning that they depend on a certain assumption being true. For instance, if an online retail system provides its own credit card payment-processing functionality, then stronger security (e.g., secure HTTP connections, two-factor authentication) is more important than low operating cost. This preference might be reversed if payment processing were outsourced to a third party.

Preferences among possible requirements for a software system may also involve comparisons between *sets of items*, rather than individual items. As an example, hospital administrators acquiring a medical records management system would likely prefer to maximize both system reliability and patient data confidentiality, compared to the less-important goals of minimizing operating cost and maximizing ease of use. However, relationships among these individual goals may not be clear from this broader statement, or even consistent with it. Yet many widely used techniques for analyzing preferences (e.g., [1]) are limited to quantitative modeling of unconditional tradeoffs between pairs of individual items. Although such techniques are useful for many purposes, they may obscure the nuances of preferences among larger sets of items, which play an important role in requirements engineering.

Conditional importance networks (*CI-nets*) [2] are a recently developed formalism that allows users to accurately model and reason with a wider variety of qualitative preferences than allowed by other formalisms (namely, conditional preferences over sets of one or more items). CI-net statements consist of four sets arranged in the following form:

$$\{\textit{true-conditions}\},\ \{\textit{false-conditions}\}:\ \{\textit{more-preferred-items}\} \succ \{\textit{less-preferred-items}\}$$

The first two sets may be empty, while the remaining two sets must each contain one or more items.

The preferences in the previous examples can be represented naturally as CI-net statements without loss of information. The first example can be written as the CI-net statement

$$\{\textit{In-House Card Processing}\},\ \{\}:\ \{\textit{High Security}\} \succ \{\textit{Low Operating Cost}\}$$

Note that the preference $\{\textit{High Security}\} \succ \{\textit{Low Operating Cost}\}$ holds only if the condition $\{\textit{In-House Card Processing}\}$ is true. (See Section II-B for more information about CI-nets.)

The main objective of our work is to improve requirements modeling, analysis, and negotiation by making it easier to incorporate set-based conditional preferences, modeled as CI-nets, within these processes. We believe that CI-nets can play a vital role in solving the following problems, even when dealing with large sets of preferences expressed by many stakeholders:

1) Identify where certain requirement preferences conflict with one another, or where preferences are incompatible because of inherent tradeoffs in the system design.
2) Choose the "best" set of requirements, i.e., the set of requirements that best aligns with the preferences of the largest and/or most important collection of stakeholders.
3) Help system stakeholders visualize the set of requirement preferences, as well as any conflicts between preferences, in ways that help them comprehend the space of possible system design choices and guide them toward efficiently resolving conflicts between preferences.
4) Support traceability between preferences and any requirements that are derived from them, and document the evolution of the preference model as stakeholders negotiate compromises and as new requirements and preferences come to light.

We aim to apply these techniques in the context of the Goal-Oriented Requirements Engineering (GORE) methodology

978-1-4673-6905-3/15 © 2015 IEEE

RE 2015, Ottawa, ON, Canada
RE:Next! Track

Accepted for publication by IEEE. © 2015 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

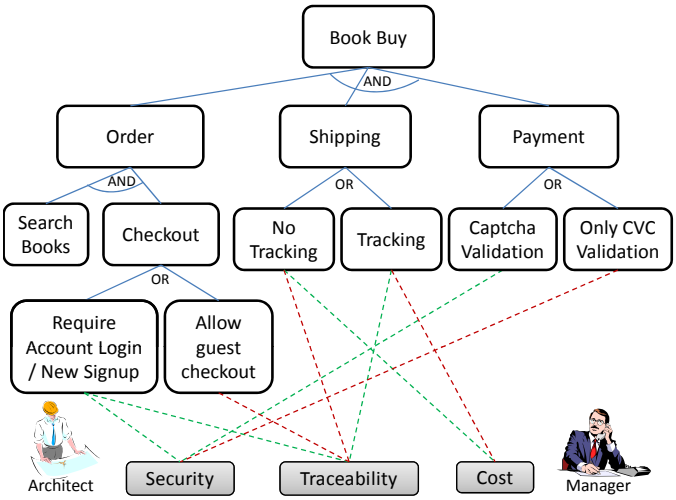

Fig. 1. Sample GORE goal model for online retailer

developed by Mylopoulos et al. [3]. In GORE, requirements for a system are represented as a *goal model*, which illustrates the AND-OR decomposition of a *root goal* into more concrete lower-level goals for the system. *Softgoals* and/or *optional goals*, which may be linked to the main goal model, specify properties that are desirable but are not part of the system's core requirement set. Several other researchers, including [4] and [5], have proposed methods for prioritizing possible system implementations according to preferences among softgoals and/or optional goals. Although our framework shares their goal of automating analysis of both preferences and goal models to help stakeholders select highly preferred system design options, our use of CI-nets for preferences provides support for conditional and set-based qualitative preference statements, which (to our knowledge) are not natively supported by previous approaches.

The remainder of this paper is organized as follows. Section II-A provides an overview of the GORE methodology and the need to account for preferences within the goal models used in GORE. Section II-B describes the CI-net formalism and provides an example of its use in the context of the GORE methodology. Section III summarizes our group's current work toward improving the usefulness of CI-nets for requirements modeling and preference analysis within GORE. Finally, Section IV describes our long-term vision for the impact of this work, along with some possible opportunities for collaboration with others in the RE community.

II. Background

We begin with an overview of essential concepts used in our work. After introducing the core ideas and terminology of the GORE methodology, including a tour of a sample goal model, we briefly introduce preference modeling and reasoning using CI-nets and their induced preference graphs (IPGs).

A. Goal-Oriented Requirements Engineering

In GORE, requirements for a given system are represented as a *goal model*, which illustrates the AND-OR decomposition of a *root goal* into lower-level goals for the system. The root goal expresses the overall requirement for the system, while the lower-level goals specify aspects of the root in greater detail, including possible alternatives for satisfying a higher-level requirement. Given a goal model for a proposed system, correct implementations of the proposed system can be identified by computing satisfiable assignments to the leaf nodes of the goal model [6].

Optionally, the goal model may also contain a set of *softgoals* or *optional goals*, which are not part of the system's core requirement set but are desirable properties for the system to possess. Because softgoals and/or optional goals are linked to goals within the AND-OR graph of the goal model, two different implementations of the system may support or interfere with different sets of softgoals or optional goals, even though they both fulfill the core requirements for the system. If preferences over the possible combinations of softgoals or optional goals are known, then possible implementations of the system can be prioritized according to these preferences [5].

Consider a goal model for an online retailer (Figure 1), where each core function may be realized in multiple ways. The unshaded node at the root of the AND-OR tree represents the required overall functionality of the system: in this case, the system's main purpose is to allow users to buy books. The rest of the unshaded nodes represent sub-goals, which specify parts of the functionality needed to accomplish a higher-level goal (AND-decomposition) or alternative methods for fulfilling a higher-level goal (OR-decomposition). The shaded nodes represent non-functional properties (NFPs) that the system could possess, while the MAKE (green) edges and BREAK (red) edges indicate whether a sub-goal contributes to the satisfaction of an NFP or violates it, respectively. Observe that there are many correct designs, which differ in terms of their NFPs: security (S), transaction costs (C) and traceability (T).

Let the manager and the software architect be the two major stakeholders who influence the design decisions in the project. Suppose that the manager holds preference P1, "Cost is more important than Traceability, provided the design is Secure". Further, suppose the architect holds preference P2, "Traceability is more important than Security". Any design satisfying all three NFPs is trivially the most preferred design that is consistent with the preferences of both the manager and the architect. However, such a design is not realizable because all three NFPs cannot be simultaneously optimized.

The next best alternative would be a design that satisfies two of the NFPs. Among the pairs of NFPs, we can infer that (T, C) is preferred to (S, C) because of the architect's preference (P2), and (S, C) is preferred to (S, T) because of the manager's preference (P1). Because the goal model indicates that no correct design exists that satisfies (T, C), we next explore the existence of a correct design that satisfies (S, C). Such a design exists, according to the goal model; moreover, it is provably the most preferred correct design with respect to the preferences of the architect and manager.

Although it is easy to identify the most preferred correct design manually for this small example, this task becomes

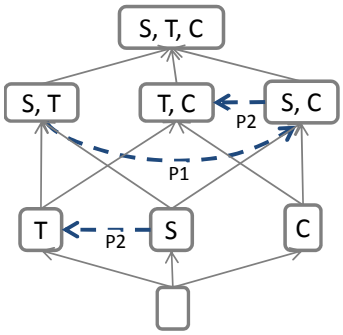

Fig. 2. Induced preference graph for CI-net with preferences P1 and P2

far more complicated in the presence of a large number of stakeholders and goal models with many nodes. Moreover, there may arise conflicts among the preferences of the various stakeholders: for example, the manager may specify that Cost is more important than Traceability and Security, but the architect may specify that Security is more important than Traceability and Cost. What is the most preferred combination of NFPs to be satisfied in this case? How would the solution change if the manager's preferences take precedence over the architect's preferences due to the manager's higher authority?

B. Conditional Importance Networks (CI-Nets)

Conditional importance networks (CI-nets) [2] provide a formalized way to specify relative importance among sets of items. We will limit ourselves here to a high-level overview of CI-nets and how they can be used; those who are interested in the details of CI-net semantics should consult [2] or [7].

Suppose a group of system stakeholders is asked to decide which of several non-functional properties are most important for a system to provide; let V be the set of all properties being considered. A CI-net $\mathbf{C}$ is a set of preference statements, where each statement contains four sets of items selected from V that are arranged as follows:

$$\{\textit{true-conditions}\},\ \{\textit{false-conditions}\}\ :\ \{\textit{more-preferred-items}\} \succ \{\textit{less-preferred-items}\}$$

All four sets must be *disjoint*, i.e., no two sets may contain the same item. The *true-conditions* and *false-conditions* sets may be empty, but *more-preferred-items* and *less-preferred-items* must contain at least one element. Informally, a CI-net statement means: "Given two sets of items, if all of the *true-conditions* are included in both sets and none of the *false-conditions* are included in either set, then the set that includes all of the *more-preferred-items* is preferred to the set that includes all of the *less-preferred-items*, all else being equal."

Automated preference reasoning over a CI-net is possible because every CI-net induces a strict partial order over the set of all possible combinations of properties in V (i.e., the powerset of V). This partial order is defined by combining the *importance rules* specified by the CI-net statements with a *monotonicity rule*, which provides an intuitive default preference in the absence of specific guidance. Informally, the monotonicity rule states that for any two sets of items A and B, if B contains every item in A plus at least one additional item (i.e., if $B \supset A$), then B must be preferred to A ($B \succ A$). As an example of the intuition behind the monotonicity rule, if offered a choice between a new car with minimal fuel included or the same new car with a full tank of fuel (at the same price), most people would choose the car with more fuel included.

Using the CI-net's preference statements and the monotonicity rule, it is possible to construct a graph where each node represents a different set of properties selected from V and each directed edge from one node to another indicates that the set of properties at the destination node is preferred to the set of properties at the source node. Such a graph is referred to as the CI-net's *induced preference graph* (IPG) [2]. An algorithm for constructing the IPG of a CI-net is described in [8]. Figure 2 shows the IPG for the example preferences given in Section II-A, which can be expressed by the following CI-net statements:

(P1) $\{S\}, \{\} : \{C\}, \{T\}$
(P2) $\{\}, \{\} : \{T\}, \{S\}$

We discuss in Section III-B our prior work toward integrating CI-nets into the context of the GORE framework, as well as our ongoing extensions of that work. In addition, we have applied CI-nets to model, analyze, and reason with preferences in a range of other problem domains, including:

1) *Trust negotiation* [8]: Given several sets of credentials that can be used to authenticate a client to a server, what is the least sensitive set of credentials (with respect to the client's privacy preferences) that is sufficient to convince the server to grant access to the client?
2) *Cybersecurity* [9]: How can a computer system administrator select the best set of countermeasures to deploy in response to an attack on the system to most effectively thwart the attacker's expected goals while causing minimal disruption to authorized users?
3) *Sustainable building design* [10]: Given the tradeoffs between the costs and benefits of sustainable building features or practices in a construction project, which collection of features and/or practices will satisfy the sustainability preferences of the client at the lowest cost?

III. Current Work

Our main objective in applying CI-nets to preference reasoning within goal models is to make set-based qualitative conditional preferences more easily usable for requirements modeling, analysis, and negotiation. We are currently pursuing two primary research directions to accomplish this objective. On one hand, we are developing ways to improve the scalability of preference analysis by developing more efficient algorithms for identifying conflicting preferences within CI-nets and for finding the most preferred sets of items as indicated by a CI-net. On the other hand, we are developing tools and related algorithms to support human users in comprehending, refining, and applying the large amount of information embedded in both CI-nets and goal models. Such tool support is especially important to make the analysis of set-based qualitative preferences

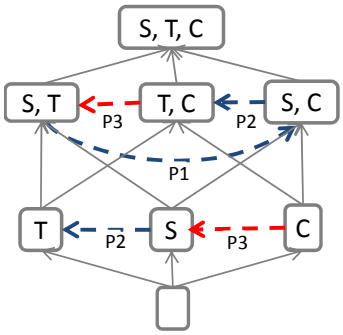

Fig. 3. Induced preference graph for CI-net with preferences P1, P2, and P3

practical for industrial-scale systems, because such preferences are not easily reduced to quickly understood (but possibly misunderstood) metrics and because requirements negotiation for a large system can be complex and time-consuming.

A. *Efficient Conflict Identification*

Before attempting to draw conclusions from a set of preferences, it is necessary that the preferences be *consistent* with each other, meaning that no preference contradicts any other preference in the set. If conflicting preferences exist, preference reasoning algorithms cannot make any guarantees about the correctness of their results. More importantly, identifying conflicts among stakeholders' expressed preferences early in the requirements engineering process can guide stakeholders toward making important decisions about the purposes and design of the system before large amounts of resources are put into implementing the "wrong" system.

Revisiting our example from Section II-B, we observe that conflicts arise when stakeholders specify opposing preferences. For example, suppose that the manager adds another preference P3, "Security is more important than Traceability", to the preference set. This can be written as the following CI-net statement:

(P3) $\{\}, \{\} : \{S\}, \{C\}$

Figure 3 shows the IPG for this new set of preferences, with the additional edges/flips induced by P3 shown in red. Note that the new IPG has cycles, indicating that the combined preferences are no longer consistent. One can reason that among the sets of NFPs, (S, C) is preferred to (S, T) due to P1, which is in turn preferred to (T, C) due to P3, which is in turn preferred to (S, C) due to P2. How can such conflicts be detected and resolved?

The preferences specified in a given CI-net are consistent if and only if the IPG for the CI-net is acyclic [2]. In [7], we used model checking to perform path-based reachability analysis on IPGs; under this approach, a given CI-net is consistent if and only if all paths through the CI-net's IPG that start at the "bottom" of the IPG (where no preference items are provided) eventually reach the "top" of the IPG (where all preference variables are provided). The results of this model checking-based algorithm have been proven correct in [7] and are simple to implement using existing model checking tools. Unfortunately, this algorithm does not scale well to large CI-nets (e.g., those with more than 10 total preference items), as $\Theta(2^N)$ time is required to construct the explicit IPG before the reachability analysis can begin. Another approach for verifying consistency among preferences would be to use an optimal algorithm for finding strongly-connected components in a graph, such as Tarjan's algorithm [11]. In this case, absence of strongly-connected components indicates a consistent set of preferences. However, although Tarjan's algorithm runs in linear time, $\Theta(2^N)$ time is still needed to construct the IPG to be searched for strongly connected components.

To improve the scalability of consistency checking for CI-nets, we are exploring a "syntactic" approach to consistency checking that will correctly report any conflicts between preferences without explicitly constructing the full IPG. This approach is based on our observation that certain patterns within CI-net statements give rise to cycles in the corresponding IPG. For example, the CI-net statement

$\{\}, \{\} : \{\textit{Low Operating Cost}\}$
$\succ \{\textit{Low Operating Cost, High Throughput}\}$

contradicts the monotonicity preference in a CI-net, which implies that {*Low Operating Cost, High Throughput*} is preferred to {*Low Operating Cost*}. Given CI-net statements of the form shown in Section II-B, conflicts of this type can be found by simply examining each CI-net statement to see whether the *less-preferred-items* set includes every item in the *more-preferred-items* set; examining the entire IPG is not necessary. We are working to identify more complex patterns within either individual CI-net statements or sets of statements that, if present, would indicate either the presence or absence of cycles in the corresponding induced preference graph. Ideally, this will allow us to identify all conflicts between specific preference statements in a given CI-net without incurring the substantial overhead needed to construct the induced preference graph. Even if this is not possible, we anticipate that we will be able to use limited syntactic analysis of CI-net preferences to improve the scalability of consistency checking by (a) identifying classes of CI-nets whose preferences can be efficiently verified as consistent or inconsistent and (b) identifying the parts of an IPG that must be constructed and analyzed to decide consistency, thus eliminating the need to construct the full IPG.

Scalable techniques for consistency checking are clearly needed to enable automated reasoning over large, complex sets of preferences. However, our vision for this aspect of our work goes beyond this immediate objective. We believe that scalable CI-net consistency checking can and should be used to improve system stakeholders' awareness of conflicts and tradeoffs among competing preferences for the system. If stakeholders are aware of these conflicts and tradeoffs, which are often not apparent until the first prototypes of the system are built, they can find mutually acceptable solutions to these problems earlier in the system development cycle, saving significant amounts of time, money, and stress.

B. Efficient Identification of Preferred Alternatives

Once conflicts are identified and removed from the set of preferences, it becomes possible to identify the most preferred sets of properties that the system could have. Knowing this information allows stakeholders to identify possible implementations of the desired system that provide the most preferred sets of properties. If no such implementation exists, it should be possible for stakeholders to obtain an approximately equally preferred or next-most-preferred set of system properties. Ideally, reasoning over qualitative preferences for various properties of the system would go hand-in-hand with analysis of the possible configurations or implementation of the system specified in a goal model within the GORE methodology. As with consistency checking, though, this family of problems is computationally difficult.

The problem of deciding whether one set of items is preferred to another is known as *dominance testing*. Although the dominance testing problem is PSPACE-complete in the general case [2], we have presented in [8] a model checking-based approach for CI-net dominance testing that identifies the most-preferred set of system properties (CI-net variables) in a reasonable amount of time and space (less than one second and 7 MB of memory) for relatively small CI-nets (up to 16 statements and 10 variables) on a standard Windows laptop. Using this approach, the costs of finding the second-most preferred, third-most preferred, etc. sets of properties are about the same as the cost of finding the most preferred set of properties. The approach is based on modeling the IPG for a CI-net as a Kripke structure, using model checking to verify the Kripke structure against specially formed temporal-logic properties, and gradually constructing a weak total order over all possible combinations of system properties (CI-net preference items). The weak total order produced by this analysis is consistent with the partial order induced over sets of system properties by the IPG. In [12], we illustrated how this preference reasoning framework can be connected to analysis of goal models in the GORE methodology to identify promising implementations of a desired system.

As with checking the consistency of a set of preferences, scalability is still a major challenge when it comes to identifying highly preferred sets of system properties and for identifying feasible implementations of a proposed system that best satisfy the preferences of the system's stakeholders. Our group's current work in this area involves exploring several possible avenues for more efficient identification of preferred alternatives. One possible approach involves representing preferences using alternative formalisms, such as binary decision diagrams (BDDs), which have been used in efficient algorithms for solving related problems in model checking. Since possible implementations of a system can be identified as satisfiable assignments to a goal models [6], we are also interested in seeing how the latest advances in algorithms and tools like satisfiability (SAT) and satisfiability modulo theorem (SMT) solvers might be applied to prioritize possible system implementations in descending order of preference, as well as the role that tools from the Knowledge Representation and Reasoning community as described in [13] may be able to play. Additionally, our group is working to identify heuristics for finding approximately optimal system implementations much more efficiently than exhaustive methods might allow. The objective of this research direction is to discover creative algorithmic solutions to analyze CI-nets and their corresponding goal models in a reasonable amount of time and space, even for industry-scale systems and their large, complex requirement and preference models.

C. Improved Comprehension of Preference Information

One of the core problems in requirements engineering is ensuring traceability between requirements and the information from which those requirements were derived. In the words of Pinheiro and Goguen in [14], "Requirements traceability refers to the ability to define, capture and follow the traces left by requirements on other elements of the software development environment and the trace left by those elements on requirements." Clearly, many system requirements are affected by the preferences of the system's stakeholders, so it is important to provide traceability between preferences and their corresponding requirements. However, a requirements engineering process for even a relatively small, simple system can produce a large amount of information, and this problem is only heightened by the addition of detailed preference information to the process. Even with formalisms such as goal models and CI-nets that provide a basic organizational structure for this information, there may be too much information for a requirements specialist, much less an ordinary system stakeholder, to process at a glance.

Our past work in this area includes the iPref-R framework, which provides tool support to assist end users with preference analysis using CI-nets. iPref-R, which is available at http://fmg.cs.iastate.edu/project-pages/GUI-iPref-R/, currently provides a graphical front-end for analysis of CI-net and [T]CP-net preferences using our model checking-based approaches. The iPref-R front-end guides users through the process of creating a CI-net while hiding the complexity of the underlying preference language, then assists users in checking their specified preferences for consistency and identifying the most preferred combinations of properties. Although the current GUI for iPref-R is helpful, it falls short of the ideal in that it does not allow users to visualize the space of possible system designs or interactively explore conflicts between preferences.

One of the long-term goals of our research in this area is the development of tools and techniques for visualizing preference and requirement data in ways that allow stakeholders to see interactively what tradeoffs exist; which stakeholders expressed which preferences; or how various requirements, properties, and preferences depend on one another. Our current work in this area involves developing an easily understood visualization of the IPG induced by a given CI-net. This visualization will display basic information about preferences among possible system properties, but users will also be able to interactively query the visualization to find out more detailed information

that traces preferences to specific stakeholders or to items in the set of system requirements. Users of this visualization will also be able to discover the tradeoffs that are represented in the goal model and the preference model, which will allow them to explore how these tradeoffs affect the space of possible system design choices.

Our eventual vision for this research direction is to produce a well-designed, easy-to-use "requirement model dashboard" that allows requirements engineers and stakeholders alike to comprehend the evolving requirement base, along with the constraints, preferences, and tradeoffs that influenced the choice of requirements used to implement the system, each traceable to a specific source.

IV. Conclusion and Future Directions

We are working to improve support for the use of CI-nets in modeling preferences among desirable properties of a proposed system and to assist stakeholders in choosing implementation options that will produce a system that is more preferred to more stakeholders than any other alternative. To support this vision, we are pursuing scalable solutions to several problems that stand in the way of CI-nets being used more widely for preference modeling and analysis: identifying conflicts between preferences for the system, choosing a system implementation (set of requirements) that allows the system to provide the most preferred possible set of properties, helping system stakeholders comprehend and refine the tradeoffs and preferences that limit choices for implementing the system, and supporting improved traceability as the preferences and requirements for the system evolve together through the requirements engineering process.

In the next two years, we anticipate substantial progress in our current work toward developing more efficient techniques for preference analysis and requirement selection using CI-nets. These new techniques for CI-nets will complement or improve upon existing preference reasoning techniques based on model checking, SAT/SMT solving, and/or AI planning. We are looking forward to integrating CI-net preference modeling and analysis more fully into the GORE methodology; in fact, we hope to make this the subject of a full-length paper to be submitted to a future RE conference. However, one challenge for our group is a lack of "real-life" requirements data against which to evaluate our ideas. We welcome opportunities to collaborate with industry partners or other researchers in this area; we are interested to see how our ideas can move others' work forward.

We are just beginning to develop tools for visualizing and comprehending goal models and CI-net preference models. A prototype tool for visualizing the IPG of a CI-net should be ready within the next few months. Although this functionality is relatively simple, it will serve as the foundation for our work in visualization, comprehension, and traceability of preferences and their corresponding requirements. Eventually, we intend to combine this visualization tool with our existing graphical front-end for specifying and modifying preferences to create a "preference workbench" for specification, visualization, and editing of preferences by system stakeholders with no special expertise in preference modeling. Again, we would like to collaborate with partners in industry or academia who would benefit from such a workbench for preference reasoning.

We believe that improving support for set-based qualitative preferences in requirements engineering may greatly improve stakeholders' understanding of the choices that go into the design of a system, as well as their sense of control over those choices. As a result, systems produced by considering such preferences during the requirements engineering process will satisfy their stakeholders' preferences better than other systems, increasing stakeholder satisfaction while reducing time and other resources spent on mid-project redesigns.

Acknowledgment

This work was supported in part by U.S. National Science Foundation grant CCF 1143734.

References

[1] J. Karlsson and K. Ryan, "A cost-value approach for prioritizing requirements," *IEEE Software*, vol. 14, no. 5, pp. 67–74, 1997. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/52.605933

[2] S. Bouveret, U. Endriss, and J. Lang, "Conditional importance networks: A graphical language for representing ordinal, monotonic preferences over sets of goods," in *IJCAI*, C. Boutilier, Ed., 2009, pp. 67–72.

[3] E. S. K. Yu and J. Mylopoulos, "Understanding 'why' in software process modelling, analysis, and design," in *ICSE*, 1994, pp. 159–168.

[4] N. A. Ernst, J. Mylopoulos, A. Borgida, and I. Jureta, "Reasoning with optional and preferred requirements," in *ER*, ser. Lecture Notes in Computer Science, J. Parsons, M. Saeki, P. Shoval, C. C. Woo, and Y. Wand, Eds., vol. 6412. Springer, 2010, pp. 118–131.

[5] S. Liaskos, S. A. McIlraith, S. Sohrabi, and J. Mylopoulos, "Integrating preferences into goal models for requirements engineering," in *RE*. IEEE Computer Society, 2010, pp. 135–144.

[6] R. Sebastiani, P. Giorgini, and J. Mylopoulos, "Simple and minimum-cost satisfiability for goal models," in *CAiSE*, 2004, pp. 20–35.

[7] G. R. Santhanam, S. Basu, and V. Honavar, "Dominance testing via model checking," in *AAAI*. AAAI Press, 2010, pp. 357–362.

[8] Z. J. Oster, G. R. Santhanam, S. Basu, and V. Honavar, "Model checking of qualitative sensitivity preferences to minimize credential disclosure," in *FACS*, ser. Lecture Notes in Computer Science, C. Pasareanu and G. Salaün, Eds., vol. 7684. Springer, 2012, pp. 205–219.

[9] G. R. Santhanam, Z. J. Oster, and S. Basu, "Identifying a preferred countermeasure strategy for attack graphs," in *Proceedings of the Eighth Annual Cyber Security and Information Intelligence Research Workshop*, ser. CSIIRW '13. New York, NY, USA: ACM, 2013, pp. 11:1–11:4. [Online]. Available: http://doi.acm.org/10.1145/2459976.2459988

[10] G. R. Santhanam, S. Basu, and V. Honavar, "Identifying sustainable designs using preferences over sustainability attributes," in *Artificial Intelligence and Sustainable Design, Papers from the 2011 AAAI Spring Symposium, Technical Report SS-11-02, Stanford, California, USA, March 21-23, 2011*. AAAI, 2011. [Online]. Available: http://www.aaai.org/ocs/index.php/SSS/SSS11/paper/view/2461

[11] R. E. Tarjan, "Depth-first search and linear graph algorithms," *SIAM J. Comput.*, vol. 1, no. 2, pp. 146–160, 1972. [Online]. Available: http://dx.doi.org/10.1137/0201010

[12] Z. J. Oster, G. R. Santhanam, and S. Basu, "Automating analysis of qualitative preferences in goal-oriented requirements engineering," in *ASE*, P. Alexander, C. S. Pasareanu, and J. G. Hosking, Eds. IEEE, 2011, pp. 448–451.

[13] A. Borgida, J. Horkoff, and J. Mylopoulos, "Applying knowledge representation and reasoning to (simple) goal models," in *IEEE 1st International Workshop on Artificial Intelligence for Requirements Engineering, AIRE 2014, 26 August, 2014, Karlskrona, Sweden*, N. Bencomo, J. Cleland-Huang, J. Guo, and R. Harrison, Eds. IEEE, 2014, pp. 53–59. [Online]. Available: http://dx.doi.org/10.1109/AIRE.2014.6894857

[14] F. A. C. Pinheiro and J. A. Goguen, "An object-oriented tool for tracing requirements," *IEEE Software*, vol. 13, no. 2, pp. 52–64, 1996. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/52.506462