Reflective Analysis of the Syntax and Semantics of the i* Framework

Jennifer Horkoff[1],Golnaz Elahi[1], Samer Abdulhadi[2], Eric Yu[2]
[1]Department of Computer Science, University of Toronto, Canada, M5S 1A4
{jenhork, gelahi}@cs.toronto.edu
[2] Faculty of Information, University of Toronto, Canada, M5S 3G6
yu@fis.utoronto.edu, samer.abdulhadi@utoronto.ca

Abstract. Conceptual modeling notations are often designed without the benefit of empirical input. Reflective analysis of modeling languages can help find the gap between the intended design of the language and its use in practice. In this paper, we study instances of the i* goal and agent-oriented Framework to analyze differences between the core i* syntax developed at the University of Toronto and existing variations. We have surveyed 15 student assignments and 15 academic papers and presentations in order to capture and analyze the most common i* syntax variations. Through this analysis we offer insights into i* syntax and suggestions to improve the framework and increase consistency between models.

Keywords: Goal Orientation, i* Modeling Framework, Language Variation.

1 Introduction

In system development, modeling languages are introduced to serve various purposes, such as facilitating communication, making implicit information explicit, or acting as a repository for knowledge. As a modeling language is adopted and used, it is useful to actively reflect on the syntax and semantics of the language, including the original intention of the language and how it is actually used in practice, in order to inform the community of users and make refinements to the language.

In this study, we focus on the i* Conceptual Modeling Framework, a goal and agent-oriented framework introduced in [1]. The Framework is intended to capture the desires which motivate system development from an agent-oriented point of view, capturing the goals of and interactions amongst system stakeholders and the system. It is directed towards the discovery and comparison of alternative system designs at a high level of abstraction, including the assignment of alternative responsibilities and the exploration of different goal operationalizations.

Although suggestions and potential directions for use of the modeling notation have been provided [1] [2], the description of the Framework was left open to a certain degree of interpretation and adaptation. Consequently, the framework has been applied to many different areas, including requirements engineering [2], system design methodologies [3] [4], and security analysis [13] [14].

In existing work using i*, the core syntax of the i* framework has often been modified, or has evolved in different directions. Furthermore, experiences teaching the i* Framework to students has shown that they often modify the syntax presented to them, either intentionally or unintentionally. In this study, we survey various instantiations of i* models and compare these instances to the version of i* currently used at the University of Toronto (U of T) [2]. A reflective analysis of the i* Framework is performed, looking critically at our own use of the framework, and questioning the assumptions underlying syntactic choices.

In this work, we call differences between this and other styles of modeling variations. We are interested in discovering the most commonly occurred variations for both students, learning i*, and researchers, applying i* in their work. To this end, 15 student assignments and 15 academic works containing examples of i* models have been surveyed. A qualitative analysis has been performed in order to understand the motivations behind the syntax variations. In our analysis, we compare the perceived motivations behind these variations to the original motivations behind the U of T syntax. As a result of our analysis, we clarify the meaning of several commonly occurring syntactical structures, make recommendations for the use of i* syntax, and provide some suggestions for potential language modifications.

2 Related Work

Although use of graphical models and modeling processes have been widely studied in software engineering research, the design of modeling notations is largely unscientific [5]. For example, the decisions about how to graphically represent constructs are made based on personal taste and intuition rather than scientific evidence [5]. In addition, Gurr and Tourlas [6] assert that many formal accounts of diagrammatic languages confuse or destroy any natural reading of the diagrams. Moody [5] argues that most diagrams do not communicate effectively and act as a barrier rather than an aid.

State of the art research in modeling languages raises the question of how modeling languages should be designed systematically, and how the existing modeling notation should evolve methodologically? The contribution in [6] expresses the need to define intuitive and natural models to move toward principled design of modeling notations. Gurr and Tourlas assert that an “intuitive” representation is one which is well matched to what it represents. Furthermore, whether a representation is “natural” concerns how it achieves its intuitive matching.

A few research studies address common mistakes specifically in using modeling notations. Among them, Lange and Chaudron [7] report results from observations in industry that show the amount of violations of completeness and consistency rules among UML models developed in practice is very high. They also conducted an empirical study to investigate to what extent implementers detect defects and to what extent defects cause different interpretations by different readers [8].

Although defect detection is not a concern of our work, we are nevertheless interested in the effect of syntax variances on understandability and comprehension of the models. We also differ from [8] in our approach to the study. In [8], UML models with deliberate defects were presented to the subjects of the experiment; while in our approach, we overviewed students' course assignments and academic papers to discover common mistakes.

Finally, Webster et al. [9] survey several published papers to collect both good practices and misuses of the i* Framework, to help requirements engineers to use the i* Framework in its full capacity. The misuses of i* mentioned in [9] overlap with the common variations we discover in this work. However, we further analyze the mistakes to discover why users make the variations and how the modeling language could be refined according to the current practice.

To our current knowledge, no studies exist which gather common syntactical variations of a modeling notation and suggests refinements to the modeling language based on patterns of mistakes that users make in practice.

3 The i* Framework

Here, we present the i* syntax initially developed at University of Toronto [1], and evolved in work such as [2], as the baseline for the comparison of surveyed models. Focusing on this syntax does not imply that it is the default or primary syntax, in fact we aim to analyze the common variations of this syntax in order to understand its weaknesses and improve its use. The i* modeling notation includes elements such as goals, softgoals, tasks, resources, and relationship links such as decomposition, dependencies, means-ends, and contributions. The Framework consists of two main modeling components: the Strategic Dependency (SD) model and the Strategic Rationale (SR) model.

In an SD model, actors in an organizational setting are defined in terms of positions, roles, and agents. The i* modeling notation provides association links to relate actors. In addition, actors can depend on each other to achieve a goals or softgoal, perform a task, or furnish a resource. An SR model extends an SD by providing task decomposition, goal refinement, and goals contributions inside the boundaries of actors. *Task-decomposition links* provide a hierarchical description of intentional elements which accomplish a task. The *means-ends links* break down a goal into alternative tasks that achieve the goal. *Contribution links* such as *help*, *make*, *hurt*, and *break* are used to express the impact of an element on softgoals. According to this syntax, tasks can be decomposed to any element. Goals can be refined using means-end relationship, and all the elements can contribute to softgoals.

Models in i* provide a basis for goal model evaluation. Procedures such as the one described in [10] propagate the level of satisfaction and denial of elements, based on a selection of alternatives, to ensure that actors' top level goals are satisfied.

4 Survey Method and Results

In order to enhance the consistency and effectiveness of the modeling process, and provide a resource for students, a set of guidelines based on the U of T i* syntax was developed and added to the i* Guide on the i* Wiki [11]. Guidelines were inspired by finding common variations via a survey of previous assignments in graduate systems analysis courses, as well as a set of available publications, all using i*. Inspired by the findings of this unofficial survey, 15 essay-type student assignments from 2006 and 2007 and 15 academic papers and presentations were surveyed again, this time taking counts of specific variations to discover the most frequent variations. The assignments were taken from three semesters of graduate courses taught by one of the authors. All assignments were surveyed, unless the students expressed an unwillingness to participate. The papers and presentations were drawn from an introductory roadmap of i* publications selected by Yu, work appearing in the i* Workshops, and work listed in the i* Wiki (see the list of Surveyed Work in the Reference Section). The survey covered all constructs of all models in each paper and assignment. The domains covered by the surveyed models were diverse, including health care, banking, and education systems. The inclusion of both academic works and student assignments, allowed for a comparison of the types of variations made by students and newcomers to the variations made in research.

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

Table 1 lists common variations in the surveyed assignments and papers. It provides total number of variations for each category of variations for student assignments and academic papers or presentations. Although we analyzed both SD and SR models, we detected variations only in SR models. Several of the variations are explained in more detail in Section 5.

5 Analysis

By performing a qualitative analysis of the motivations behind each variation, we can analyze whether the variation represents a potential source of confusion in a model, whether the convention can be seen as a shortcut for more complex syntax, or whether it indicates issues in the i* notation used at U of T. We have grouped variations together when we believe they stem from similar motivations. The counts for each variation within a particular category are provided. Due to space restrictions, we pick out only the most prominent variations to discuss.

Table 1. Summary of common variations.

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

The Nature of "Hard" Elements and Softgoals. [Decomposition links are used between Softgoals (3), Decomposition links drawn from softgoals to tasks (2), Means-Ends are drawn from goals to softgoals (2), and Mean-Ends are used between softgoals (1), Softgoal dependency is met by a goal (5), Softgoal dependency is met by a task (2), Contribution links are drawn from softgoals to tasks (4), Contribution links are drawn from Softgoals to goals (2), Contribution links are used between goals (1), Contribution links are drawn from resources to tasks (1), Decomposition links are drawn from goals to softgoals (5), Means-Ends are drawn from softgoals to goals (2), Softgoal should be goal (10), Goal should be softgoal (15), Task should be softgoal (8), Softgoal should be task (7), Total: 70].

Many of the variations can be attributed to a misunderstanding of the nature of hard and softgoals. Generally, users confuse hard and soft elements. Furthermore, several variations involved having a “hard”, non-softgoal element as a recipient of contribution links. In i*, a goal, task or resource is typically considered similarly to a functional requirement, they are concrete states, actions or entities, respectively. From this point of view, it does not make sense to say that another element can provide a qualitative contribution to these elements (either partial or sufficient). To keep the differences between hard and soft elements clear, we decompose hard elements using only AND/OR type links (Decomposition and Means-Ends) in order to ascribe clearly defined decompositions to concrete elements. We can also see that modelers occasionally use links associated with hard elements with softgoals, and that softgoals depend on hard elements in dependencies. For the first case, as the nature of a softgoal implies qualitative, “good enough” analysis, it is unlikely to be decomposable into strict AND or OR relationships, such as Means-Ends or Decomposition. Although the i* Framework does retain the use of AND and OR contribution links for softgoals, (adopted from the NFR framework [12]), their use is infrequent..

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

In several cases, modelers have decomposed goals to softgoals, violating the restrictions that goals should only be decomposed to tasks. The nature of hard goals and softgoals implies that a softgoal should not be a means to a hard goal; sufficiently accomplishing a qualitative goal, should not allow the accomplishment of a concretely defined state of the domain. However, we can observe that in i* syntax a softgoal *is* allowed to be a decomposition element of a task. This seems to contradict the notion of a task as a concrete series of actions. In fact, when this type of syntax is used, we interpret the task to represent not only the concrete actions, but also the desired qualities that this particular task should accomplish in order to be satisfied. For example, in the left snippet of Fig. 1, Send Message is only satisfied if the Message is Sent Securely. If the message is sent, but it was not secure, send message is denied. Such a situation can also be created when tasks or goals depend on softgoals.

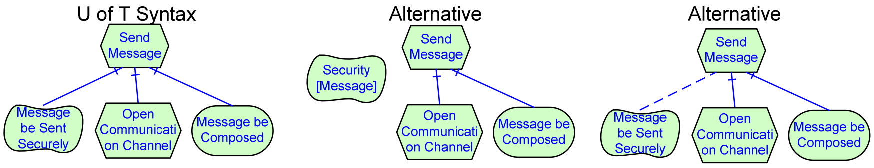

Fig. 1. Example Task Decomposition (left), Alternative Syntax Examples (middle, right)

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

Possible Responses. Although a solution to these issues may be to discontinue the decomposition of Tasks to softgoals, there remains a need to explicitly associate non-functional qualities with functional elements. In the NFR Framework [ ], this was done using a type and topic style of naming, where goals were named by the type of softgoal (security, ease of use, etc.) and their domain specific topic, as "Type [Topic]", see the middle of Fig. 1. Alternatively, a visual way to associate softgoals to functional elements which does not directly affect the evaluation of the functional elements could be devised, allowing the "hard" elements would retain their binary meaning. This alternative is shown on the right of Fig. 1. More investigation into the usability of the last option is required.

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

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

The restrictions concerning Decomposition and Means-Ends links can be justified by the notion of tasks versus goals, and by the desire to prompt for the discovery of alternatives. In [2], a goal, by definition, can be accomplished in different ways, whereas a task specifies one particular way of accomplishing something. Thus, in Fig. 2, modeling Appointment Be Scheduled as a Goal would indicate that there are several different ways to schedule an appointment, while choosing to model Financial Management as a Task indicates that this refers to one particular way of performing financial management.

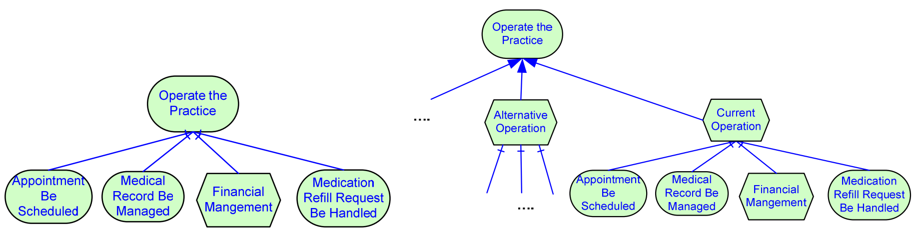

Fig. 2. Example of Decomposition Variation Recreated from [15] (left), Redrawn in the U of T Style (right)

In the U of T style, the left side of Fig. 2 would be redrawn as shown on the right. By adding the extra task (Current Operation) between the goal decompositions and the original goal, we emphasize that this set of decompositions composes only a single way to decompose and accomplish the task. There can, in fact, be several ways to decompose the high-level goal, each having potentially different effects on qualitative aspects represented as softgoals. Despite these reasons, for reasons relating to the scalability and simplicity of i* models, users often relax the rules concerning Means-Ends and Decompositions.

Possible responses. With this in mind, we propose two levels of i* syntax, a strict level which follows the syntax laid out in Section 3 and a looser level which uses syntactical shortcuts. In the strict version of the syntax, restrictions such as those concerning Means-Ends and Decomposition would apply. In the looser level, these restrictions can be relaxed, allowing users to be more concise. Therefore we can consider the left side of Fig. 2 as a "shortcut" for the right side. When a modeler chooses to use this simplified syntax, the underlying meaning, represented by the more detailed syntax, should be clear to the modeler and the model readers. If there is any doubt concerning the clarity of meaning, the stricter syntax should be used.

Actor Boundaries . [Decomposition links extend outside actors' boundaries (4), Dependency links are used inside actors (1), Means-Ends extend outside actors' boundaries (3), and Contribution links extend outside actors' boundaries (6), Total: 14]

One frequently observed variation is that a decomposition, means-end, or contribution link extends outside of an actor's boundary. In the U of T version of i* syntax, all of these instances would be replaced by dependency links. It is important to limit non-dependency links to inside boundary of the actors to emphasize on actors' autonomy. In this way, externally visible actor relationships are limited to dependency link, and other actors do no have knowledge of the inside motivations of an actor. This situation better reflects the autonomy of actors occurring in the domain.

By only using dependency links across actor boundaries, one can ensure that the SR model is consistent with the SD model, avoid confusion translated between the two. However, practitioners frequently violate these rules, and according to detail analysis of the models, scalability and usability issues lead to these variations. Although these variations are not compatible with the notion of actor autonomy, they communicate the same semantics represented with strict rules with a looser syntax which works as a shortcut. For example, Fig. 3 shows an example of a variation in [14] and its representation using the strict syntax of U of T style.

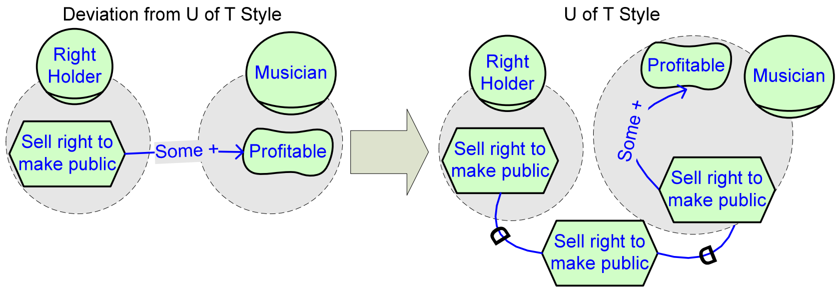

Fig. 3. An example of a variation used in a paper and its representation using the U of T style.

6 Discussion

By analyzing the results in Table 1, we can perceive several differences between the variations found in student assignments and academic work. It appears that students have more difficulty in understanding the nature of softgoals, and the differences between soft and hard elements. Although these notions are likely familiar to researchers, students are likely to be new to these ideas. Similarly, we see that students are more likely to have incomplete models, lacking softgoal decomposition, and are more likely to misuse association links. These issues can be addressed by placing greater emphasis on these concepts when teaching i* to new users.

On the other hand, we can observe that researchers are more likely to use non-dependency links outside of actor boundaries. We can postulate that researchers are more likely to adapt the Framework as they see fit. If they are faced with scalability issues, are more likely to deviate from the syntax laid out in [1] and [2]. No other significant differences between student and research results are found.

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

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

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

7 Future Work and Conclusions

Through a survey of student assignments and academic work, we have discovered frequent variations from the i* syntax developed at U of T. Through a qualitative analysis of these variations, we have attempted to understand the motivations for these variations, and compared these motivations to the original motivations for the syntax introduced by Yu [1] [2]. We suggest the use of strict and loose versions of i* syntax, the latter containing syntactical shortcuts for configurations in the former. The emphasis of this paper is not to avoid the loose syntax, but to be sure that modelers and model readers have a consistent interpretation of syntactical shortcuts. Suggestions for modifications to i* syntax involving a clearer separation of hard and soft elements have been made. Future work should involve empirical assessments of the effect of adopting the loose syntax on the resulting models and model reasoning.

The understanding facilitated by our analysis can promote consistency in i* modeling styles and model interpretation, and help to avoid ambiguous syntax. We have also pointed out potential areas of confusion for new i* users, allowing for a more effective formulation of curriculum involving i* modeling.

In the future, we would like to expand our survey to include more student assignments and academic papers in an attempt to verify or refute our results. In ongoing work, we have obtained access to i* models developed for industrial purposes in a software maintenance organization and are in the process of incorporating analysis of those models into the current study. We are also interested in examining the relationship between variations and model size, exploring to what degree variations are motivated by scalability issues. In addition, we are initiating a call to collect i* syntax variations from research groups using i* in the i* collaborative wiki. We have an "Open" version of the i* Guide, and invite researchers to contribute their varying i* syntax to the Guide [11]. By inviting others to provide examples and explanations of their syntax, we avoid the bias introduced when one group surveys the conventions of other groups. By collecting and comparing differing i* syntax, we can open a dialog concerning the motivations behind these differences, leading towards a more universal understanding of i* models.

Acknowledgments. We gratefully acknowledge financial support from the Natural Science and Engineering Research Council of Canada (NSERC) and Bell University Labs.

8 References

1. Yu, E.: Modeling Strategic Relationships for Process Reengineering. PhD thesis, Department of Computer Science, University of Toronto, Canada (1995)
2. Yu, E.: Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering. In Proc. of the 3rd Requirements Engineering Conf., RE'97, pp. 226-235. (1997)
3. Castro, J., Kolp, M., Mylopoulos, J.: A requirements-driven development methodology. In Proc. of the 13th Int. Conf. on Advanced Information Systems Engineering, CAiSE'01, pp. 108-123. (2001)
4. Jones, S. Maiden, N.A.M.: RESCUE: An integrated method for specifying requirements for complex socio-technical systems. In J. L. Mate and A. Silva (eds.), Requirements Engineering for Sociotechnical Systems, Idea Group Inc. (2005)
5. Moody, D.: What Makes a Good Diagram? Improving the Cognitive Effectiveness of Diagrams in IS Development. In Proc. of the 15th Int. Conf. in Information Systems Development, ISD'06, pp. 481-492. (2006)
6. Gurr C., Tourlas, K.: Towards the principled design of software engineering diagrams. In Proc. of the 22nd Int. Con. on Software Engineering, ICSE'00, pp 509–518 (2000)
7. Lange, C.F.J., Chaudron, M.R.V.: An Empirical Assessment of Completeness in UML Designs. In Proc. of the Int. Conf. on Empirical Assessment in Software Engineering, EASE'04, pp. 111-119. (2004)
8. Lange, C. F. J., Chaudron M. R. V.: Effects of defects in UML models: an experimental investigation. In Proc. of the 28th Int. Con. on Software Engineering, ICSE'06, pp. 401-411.

(2006)
9. Webster, I., Amaral, J., Cysneiros Filho, L. M: A Survey of Good Practices and Misuses for Modelling with i* Framework, In Proc. of the VIII Workshop on Requirements Engineering, WER'05, pp. 148-160. (2005)
10. Horkoff, J.: Using i* Models for Evaluation, Masters Thesis, University of Toronto, Department of Computer Science (2006)
11. i* Guidelines, version 3 (2007) http://istar.rwth-aachen.de/tiki-index.php?page_ref_id=67
12. Chung, L., Nixon, B. A., Yu, E., Mylopoulos, J.: Non-Functional Requirements in Software Engineering, Kluwer Academic Publishing (2000)

Surveyed Work

13. Liu, L., Yu, E., Mylopoulos, J.: Security and Privacy Requirements Analysis within a Social Setting, In IEEE Joint Int. Conf. on Requirements Engineering, RE'03, pp. 151-161. (2003)
14. Gans, G., Jarke, M., Lakemeyer, G., Schmitz, D.: Deliberation in a metadata-based modeling and simulation environment for inter-organizational networks, In Proc. of the 15th Int. Conf. on advanced information systems engineering, CAiSE'03, pp. 587-607. (2005)
15. Samavi, R., Yu, E., Topaloglou, T.: Strategic reasoning about business models: a conceptual modeling approach. In Journal of Information Systems and E-Business Management (2008)
16. Grau, G., Franch, X., Maiden, N. A. M.: A goal based round-trip method for system development. In Proc. of the 11th Int. Workshop on Requirements Engineering: Foundation For Software Quality, REFSQ, pp. 71-86. (2005)
17. Raadt, V. D., Gordijn, J., Yu, E.: Exploring Web Services from a Business Value Perspective. In IEEE Joint Int. Conf. on Requirements Engineering, RE'05, pp. 53-62. (2005)
18. Lespérance, y., Lapouchnian, A.: On Using i* for Modeling Autonomy, Reasoning, and Planning in Adaptive Systems, Presentation in istar'08 Workshop (2008)
19. Oliveira, A. P. A., Prado Leite, J. S. C., Cysneiros, L. M.: AGFL - Agent Goals from Lexicon Eliciting Multi-Agent Systems Intentionality, Presentation in istar'08 Workshop (2008)
20. Kolp M., Faulkner, S., The SKwyRL Approach: Social/Spiral Design based on i* and Tropos, Presentation in istar'05 Workshop (2005)
21. Maiden N.A.M., Kamdar N. & Bush D.: Analyzing I* System Models for Dependability Properties: The Uberlingen Accident. . In Proc. of the 12th Int. Workshop on Requirements Engineering: Foundation For Software Quality, REFSQ, (2006)
22. Elahi, G., Yu, E.: A Goal Oriented Approach for Modeling and Analyzing Security Trade-Offs. In Proc. of 26th Int. Conf. of Conceptual Modeling, ER'07, pp. 375-390. (2007)
23. Strohmaier, M., Yu, E., Horkoff, J., Aranda, J., Easterbrook, S. M.: Analyzing Knowledge Transfer Effectiveness -An Agent-Oriented Modeling Approach, In Proc. of 40th Hawaii International Conference on Systems Science, HICSS'07, pp. 188 (2007)
24. Yu, E., Strohmaier, M., Deng, X.: Exploring Intentional Modeling and Analysis for Enterprise Architecture. In Proc. of the EDOC 2006 Conf. Workshop on Trends in Enterprise Architecture Research, TEAR 2006, (2006)
25. Mazón, J., Trujillo, J., Serrano, M., Piattini, M.: Designing data warehouses: From business requirement analysis to multidimensional modeling, In Proc. of Int. Workshop on Requirements Engineering for Business Needs and IT Alignment (2005)
26. Arzdorf, t., Gans, g., Jarke, m., Lakemeyer, g., Schmitz, d., SNet: A Modeling and Simulation Environment for Inter-Organizational Networks, Presentation in istar'05 Workshop (2005)
27. M. Kolp, T.T. Do, S. Faulkner, Multi-Agent Architecture for E-Business Systems: An Organizational Perspective, In Proc. of 23rd Int. Conf. on Information Systems, (2002)