

Instructional Experiences with Modeling and Analysis using the i* Framework

Zia Babar¹, Soroosh Nalchigar², Lysanne Lessard³, Jennifer Horkoff⁴, and Eric Yu^{1,2}

¹Faculty of Information, University of Toronto, Toronto, Canada

²Department of Computer Science, University of Toronto, Toronto, Canada

³Telfer School of Management, University of Ottawa, Ottawa, Canada

⁴Department of Human Computer Interaction, City University London, London, UK

zia.babar@mail.utoronto.ca, soroosh@cs.toronto.edu,

lessard@telfer.uottawa.ca, horkoff@city.ac.uk,

eric.yu@utoronto.ca

Abstract. First year professional Master's students at the Faculty of Information, University of Toronto are taught how to analyze problem domains using i* modeling in the final segment of the introductory course on "Systems Analysis and Process Innovation". Over the past several years, the course instructors and teaching assistants have had the opportunity to observe and interact with the course students on the learning and use of the i* framework for modeling and analysis which makes it possible to make general observations regarding student experiences with the i* framework in an instructional setting. These observations are broad ranging and cover areas such as pedagogical design, conceptual understanding and proficiency with the i* framework, domain modeling and analysis, and tools for i* modeling.

Keywords: experience report, i* framework, agent-oriented modeling

1 Introduction

i* modeling [1] has been taught at a number of universities around the world for more than a decade. This workshop offers the first opportunity for i* educators to share experiences and to collectively advance pedagogical knowledge and practice. The "Systems Analysis and Process Innovation" course [2] at the Faculty of Information, University of Toronto includes i* modeling in a segment that introduces the strategic actors perspective as complementary to well-established methods of information systems analysis. In this paper we aim to stimulate discussion and help generate ideas that would improve i* pedagogy for future offerings of this course as well as for the i* teaching community at large. In an earlier work [3] a qualitative analysis was done by surveying 15 student assignments and 15 academic papers; the discussion points in this paper supplement this previous work. However, the observations in this paper are not the result of a systematic study of student assignments but rather are based on our teaching experiences and recollections as instructors and teaching assistants in this course.

2 Contextual Factors

Since teaching and learning experiences can vary widely according to contextual factors, we outline the educational setting of the course and the role of i* modeling within the course.

This course is taken primarily by first year professional Master's students at the Faculty of Information. Students come from a diversity of academic backgrounds, ranging from humanities to social science, to business, and to health sciences. A small number have prior degrees in computing or information systems. Thus for many students, the course is their first exposure to systems analysis and modeling, while at the same time they are learning basic concepts of information technology and computer programming in concurrent courses. Upon graduation, the most common job role for past students of this course is "business analyst".

The aim of the course is to introduce students to the methods, techniques, and practices of systems analysis, emphasizing the need to characterize as-is and to-be configurations of systems in organizational context at the conceptual level of "requirements" rather than at the level of implementation technology. The course also highlights the varying degrees of change that can result from systems analysis, ranging from "automation" that involves minimal reconfiguration of business processes, to "transformation" that rearranges relationships among key stakeholders, as in business model redesign.

Modeling techniques are introduced as conceptual tools to aid the representational and analytical activities of the analyst. The emphasis is not on the various languages per se, but rather on the modeling paradigms that they represent – BPMN and DFD as illustrating the process-oriented paradigm, ERD and class diagrams for the data-oriented paradigm, UML for the object-oriented paradigm, and i* as representative of a goal- and agent-oriented paradigm. The intentional strategic actors perspective of i* is further contrasted with the value network perspective of VNA [4].

An important objective is for students to acquire a critical reflective understanding of a range of modeling paradigms and associated techniques, their capabilities and limitations, as well as their underlying motivations and suitability for various contexts. To this end, students engage in real-life case studies for project assignments rather than rely on exercises and drills. The pedagogical design of the assignments is further described in a companion paper [5].

3 Experiences and Discussions

Given the above instructional context for i*, we discuss our experiences, observations, and potential avenues for improvements in several areas.

3.1 Conceptual Understanding of the Premises and Objectives underlying i*

The students were generally appreciative of the distinctive line of reasoning of the i* framework and its complementary nature to other modeling techniques. Over the

duration of the course, students progress from process and data modeling to agent-oriented modeling [6]. The core concepts of the i* framework, particularly the ability to broadly model and evaluate an organizational setting, were relatively accessible to the students. Students acknowledged the capability of the i* framework to depict a social organization which helped them visualize the various motives, beliefs, intentions and dependencies that typically exist in an organization; aspects which are generally not acknowledged or supported in other modeling techniques. They were able to critically contrast i* with VNA, and with process modeling.

3.2 Proficiency in i* Modeling

Although the course objective is not to learn the i* language (or any other specific language) per se but to appreciate and experience first-hand the strategic actors thinking expressed through i*, students nevertheless need to acquire a sufficient level of proficiency in the language in order to benefit from the experiential learning in real-life projects.

Despite the limited instructional hours allocated to the i* segment in the course, we found that students were able to construct i* Strategic Dependency (SD) models fairly readily, and Strategic Rationale (SR) models with some extra effort. The following concepts typically require more attention. Most students overcome these conceptual barriers upon clarification during class or project discussions, although misconceptions also appear in final assignment submissions.

For SD models:

- Differentiating goal dependencies from task dependencies. Similarly for goal dependencies versus soft-goal dependencies.
- The direction of the dependency relationship between “depender” and “dependee” actors is at times incorrectly indicated.
- The naming of goals and other intentional elements are often stated according to domain-specific priorities (e.g., “Make more money”) rather than in accordance with naming guidelines for each type of element (e.g., “Profits be increased” for a softgoal vs. “Increase sales revenue” for a task goal).
- Distinctions among i* actor types (roles, positions and agents) are rarely used, as are IS-A relationships among actors. These are noted as more “advanced” concepts in class, to avoid cognitive overload for beginner learners. Their usage is recommended on an as-needed basis.
- IT systems are often modeled initially as resources to be used by other actors. The benefits of treating non-humans as actors require further explanation.

For SR models, difficulties often relate to the link types:

- Decomposition is readily understood, whereas means-ends requires more effort. Initial models often confuse the two.
- A high-level goal is often directly decomposed into sub-goals without determining the possible means of achieving that end. This limits exploration of alternatives, and reduces the potential for model analysis, evaluation and alternative selection.

- Contribution links are generally understood, once the softgoal vs. goals distinction is made; however initial models often confuse means-ends with contribution links.

3.3 From Modeling to Analysis

Throughout the course, the analytical powers of models are emphasized so that they are not seen as merely descriptive. In student work, such analytical capabilities are typically under-utilized. This applies to the earlier assignments involving process modeling, as well as to the assignment where i^* is used. Models serve as descriptions, but conclusions are often drawn based on domain knowledge rather than using the models. The concepts of goal satisfaction analysis are typically not fully exploited; while students typically do perform some type of analysis to fulfill assignment requirements, they do not always use the results of these analyses to make recommendations. The reasons for this could be varied, ranging from incomplete domain models that lead to unhelpful or incorrect results, to students' inexperience with model evaluation and analysis. As a result, students may be unable to provide strong and justified explanations on how the modeling analysis exercise helped them to come up with redesigned-to-be alternatives which solve the issues in the problem domain.

3.4 Managing Model Complexity

Managing complexity and layout within i^* SD and SR diagrams takes particular care because of the flexibility of usage that the i^* framework provides. It is easy to get carried away with modeling a domain which can lead to unmanageable models. Following the layout rules for i^* models (such as no intersecting links or dependency links having a natural flowing line [7]) can be challenging in large models. Visualization of the problem domain becomes increasingly problematic with the increasing number of intentions, actors and links. Transferring a large model drawn in a software tool to an assignment document can result in an illegible figure, unless extra effort is made to develop proper views or model subsets. To help simplify the final i^* model, students were encouraged to remove elements of the model which are not important to the domain analysis. However this is often most easily done once the initial model was developed and students have determined that certain actors (and their associated intentions and links) are not important to subsequent analysis and alternative selection. Producing the final model can require multiple iterations requiring significant investment in time and effort.

3.5 Teaching Resources

Students are introduced to i^* through class lectures with detailed examples, and readings of selected representative papers on i^* . The course adopts the version of i^* as described in [8]. The i^* wiki [7] serves as a supplementary resource. It was meant as a guide for students new to the i^* framework as well as providing a reference guide to experienced modelers. Nevertheless, the availability of pedagogical literature and learning aids are not comparable to well-established modeling techniques and stand-

ardized languages (such as BPMN and UML) which have extensive literature and educational material available for students for making the progression from basic to advanced modeling and analysis more quickly.

A useful addition to the existing material would be a set of usage scenarios and examples that cover progressively more difficult cases, preferably using a large, real-world problem domain. The series of examples taken as a whole should be comprehensive so that the lesser used or more difficult to understand constructs (like IS-A relationships, different actor types, different contribution links etc.) are also covered. Detailed explanations on the various goal satisfaction analysis methods and their proper usage could then be provided by using this depicted domain model as an example setting. Instructional videos could be made which show the best practices and guidelines for i* modeling and analysis. These could be embedded within the wiki itself or uploaded to a video sharing site such as YouTube. These resources would further help students adjust to the agent- and goal-oriented paradigm in the limited time available to them and help them model and analyze problem domains using i* effectively and comprehensively. Such an endeavor could benefit from collaborative effort from the i* community.

3.6 Tools for Supporting i* Modeling, Analysis, and Model Management

Typically students in the course create i* SD and SR diagrams using either an i* stencil for Microsoft Visio [9] or OpenOME [10]. Students are usually familiar with Microsoft Visio and are able to quickly use the i* stencil to start modeling the problem domain. OpenOME is a specialized software tool developed for agent- and goal-oriented modeling and analysis. It offers and supports language-specific concepts such as syntax checking, model statistics, and goal satisfaction analysis.

Recent advances in browser side web technologies, such as HTML5 and JavaScript, have made it possible to have a lightweight i* modeling software tool that runs entirely as a web application in a browser with the main server-side application hosted on a central system (similar to, for example, LucidChart, Gliffy). Any improvements made to this deployed environment would be immediately available to everyone as everyone accesses the same server. In addition to the capabilities of OpenOME and other i* related tools, the new software tool could have capabilities such as immediate checking of syntax and best practice violations, auto layout of i* models, built-in tutorials and examples, cooperative and collaborative modeling for student groups, ability to hide/show segments of the i* model, etc. Some of these features have already been implemented in other tools. Such improvements could make analysis and modeling of the problem domain using i* framework further accessible to students.

3.7 Pedagogical Design

The assignments and studio presentations are structured to encourage reflection, comparison between multiple modeling techniques and proposal of alternative methods of achieving organizational objectives. The pedagogical design of the course is covered in some depth in a companion paper [5]. Studio presentations take place towards the

end of the course where each student team is asked to present and explain these models to their fellow students, who then provide constructive feedback. As part of this experience, student teams are able to considerably improve their domain model (in terms of depiction, syntactic use and analysis) before their final assignment submission. These improvements could be traced back not only to fellow students' suggestions, but also to insights gained as a result of seeing or commenting on other students' work.

4 Conclusions and Future Work

The intention of this paper was to review and document our instructional experiences as course instructors and teaching assistants for the "Systems Analysis and Process Innovation" course. By sharing our experiences with the wider i* community, we hope to be able to collectively further develop instructional methods and enhance the in-class learning experience across the many universities where similar courses are on offer. Continuous pedagogical improvements to the course can benefit from systematic analysis of submitted assignments, subject to research ethics protocols. Input from students on specific aspects of the i* framework, through surveys or other means, can also be sought.

5 References

1. Yu, E., Giorgini, P., Maiden, N., Mylopoulos, J.: *Social Modeling for Requirements Engineering*. MIT Press (2011)
2. "Systems Analysis and Process Innovation" course outline, <http://yu.ischool.utoronto.ca/1341outline.pdf>
3. Horkoff, J., Elahi, G., Abdulhadi, S., Yu, E.: Reflective Analysis of the Syntax and Semantics of the i* Framework. In *Advances in Conceptual Modeling—Challenges and Opportunities*, pp. 249-260. Springer Berlin Heidelberg (2008)
4. Allee, V.: Value network analysis and value conversion of tangible and intangible assets. *Journal of Intellectual Capital*. 9(1), 5-24 (2008)
5. Yu, E., Lessard, L., Babar, Z., Nalchigar, S., Horkoff, J.: Teaching i* Alongside a Contrasting Modeling Framework. Submitted to iStarT'15. (accepted)
6. Yu, E.: Why Agent-Oriented Requirements Engineering. In *Proceedings of 3rd Workshop on Requirements Engineering For Software Quality*. (1997)
7. i* Wiki, <http://istar.rwth-aachen.de/>
8. Yu, Eric S.K.: Towards modelling and reasoning support for early-phase requirements engineering. In *Requirements Engineering, Proceedings of the Third IEEE International Symposium on*, pp. 226-235. IEEE (1997)
9. MS Visio i* stencil, <http://istar.rwth-aachen.de/tiki-index.php?page=VISIO>
10. OpenOME, <https://se.cs.toronto.edu/trac/ome/wiki>