From *i** to *iStar 2.0*: An Evolving Social Modelling Language

Lin Liu

[1] School of Software, Tsinghua University, Beijing, 100084, China
linliu@tsinghua.edu.cn

Abstract. Conceptual Modelling, as a thought tool, helps its adopters to describe an abstract observation to given real world phenomena. *i**, as a social modelling language, was widely adopted by researchers in both requirements engineering and business information system analysis. In order to further extend its adoption in future research and practice, *iStar* 2.0 was conceived and published to reduce ambiguities and complexities. In this paper, I would like to share my observations as an *i** modeler, about the major differences identified between *i** and *iStar* 2.0, about how to map an *i** model to an *iStar* 2.0 model, as well as how to further evolve the modelling language to serve for the next generation modelling needs.

Keywords: social modelling, concepts, relations, evolution

1 Motivation

My first contact with *i** was in year 1999, a few months before my graduation and joining Eric's group, I downloaded two of his major *i** modelling papers, in which he used *i** to analyse business process of IKEA [1], and the mutual dependencies between members in software project team [2]. At that time, my world model was already agent-oriented, due to my master's thesis, on Actor [3], the concurrent computing model, and PhD thesis, on Agent-oriented requirements analysis using Agent-Z [4] and process algebra to formalise use case scenarios. My first impression to the example models was: it is a bit complex, and drawn artistically, different from most other software engineering models that I was familiar with, such as, message sequence charts, state machines, and class diagrams.

Tonight, I finally get a chance to read the *i** 2.0 Language Guide [5] by Fabiano, Xavier and Jennifer, word by word, which I should have done two and half years earlier, while they sent numerous messages asking for comments and inputs. I feel synchronised with the evolving language finally, alas, it is better late than never.

2 Related work – Bits and pieces in Retrospect of 17 Years personal modelling and extension experience with *i**

2.1 A Successful exercise - Modelling Trust in Smart Cards

It didn’t take me long to falling in love with using *i** to model real world. As my first modelling exercise was a perfect match. Modelling trust in smart-cards systems can make use of many of the interesting modelling constructs in *i**[6]. Including:

Strategic dependency modelling captures the major roles in a general smart-card system, and their dependencies.

Role, position and agent together with role-playing and position occupancy links, which are used to capture who is playing attacker, and which organisational player is playing the abstract role of card owner, software owner, data producer, and how the trust and dependency distributed among different organisational settings.

The flexible use of contribution links, in particular, the attacks are represented as a “**break**” contribution ignited from the attacker to the victim or vulnerable element or link.

2.2 A first attempt to extend *i**– capturing temporal orders between tasks with prior-to link

My second modelling exercise was due to the marriage of GRL with UCM [7]. GRL is a variation of *i**, which emphasis more the goal-oriented perspectives of *i**, I think I tried to combine the two language in a different way comparing to what the current URN standards suggested. I wanted to use *i** model as the container, or the place-holder of scenarios. I only wanted the joggle line of UCM, which traversing the tasks in *i** played by different actors. It captures the refining process of goals and generated the run-time execution scenarios involving multiple actors in the model. In my mind, goals, actors, and scenarios are the three pieces to be fit together in the requirements engineering puzzle.

2.3 Another successful exercise – modelling service relationships

My next major move was using *i** in services modelling. Again, *i**, especially, the social, strategic dependency modelling construct, was a natural fit to model service providers and service consumers, their needs and capabilities, their delegation of different types, their commitment and delivery of services, and quality of service [8]. *i** is expressive enough support my modelling objectives and provide meaningful types of analysis.

2.4 A second attempt to extend *i**– capturing context as annotations

My second attempt to extend *i** was due to the modelling needs of service adaptation. Where I wanted to say that when service text changes, service provider and consumer’s choices will change accordingly, so I associated context information on each of the service goal refinement link [9].

I am doing this retrospect just to take a position of modeller, to see if the current iStar 2.0 concepts and relations will still allow me to do similar things in a clearer way, or it has decided to avoid such use or extension of the modelling language.

3 Redrawing *i** models with iStar 2.0

3.1 Discard organizational positions together with occupancy and coverage link

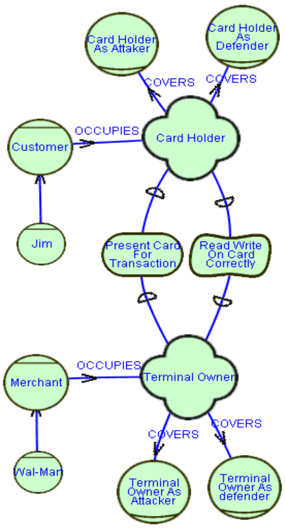

Fig. 1 actors and association links in *i** [6]

As shown on the left-hand side, in *i**, position is used to capture organizational positions, which can cover different abstract roles, and being occupied by different agents. As *i** was designed as an organizational modelling language, so position is considered as a first-class modelling concept. In iStar 2.0, organization is not stressed anymore, so we can define it as an abstract role, participated-by agents, and it can specialize either an attacker or a defender or both using is-a link. In this case, a cover link will become an “is-a”, an “occupies” will become a “participates-in”. It is not clear whether the “INS” link is represented now, as it is a relation between actors of same nature. Although I can make the mapping, I feel that it is a more natural and direct reflection of the real world meaning using “plays” and “is-part-of” than “participates-in” in this case.

3.2 Changing means-ends and decomposition as AND/OR refinement

In iStar 2.0, the “means-ends” and “decomposition” links in *i** are unified to be called AND/OR refinement links, which loosens up the original *i**’s strict enforcement on iterative elicitation of alternative ways to satisfy a given goal using “means-ends” link. As a modeller, I had hated the constraint of imposed by this “means-ends” semantics, as I had to use some dummy tasks and goals when there are not meaningful alternatives to choose from. However, I have to admit that we should keep it in mind to ask the question each time we face a goal. In other words, I would rather see the current iStar 2.0 treatment as a simplification of the *i** syntax rather than a change of semantics.

3.3 Adding Qualification link between a quality attributes with its associated elements

A new kind of link, qualification link, represented as dotted line, links the quality attributes with the corresponding goal, resource and task elements explicitly. This is a major extension in iStar 2.0, which can only be represented using the naming convention of "subject[object]" in softgoals in *i** or NFR framework [10]. This extension clarifies the semantics of quality attributes and suggests a proper way of its usage.

3.4 Adding DependerElmt, dependeeElmt, rules and constraints on social dependencies

There is a rules and constraints section in the specification of dependency relationship in iStar 2.0. It gives clearer guidelines to modellers which encourage the proper use of the link. This include: (1) adding definition of dependerElmt, dependeeElmt, and confine depender and dependee as actors; (2) when a depender delegates a dependerElmt to others, it cannot be refined or contributed by other elements within the actor boundary. Dependency relationships are not allowed to share the same name, which means there is exactly one depender and one dependee for each dependums. This extension is also a very good move in making the semantics of dependency link clear and easy to use.

3.5 Adding NeededBy relation between a task and its required resource

Resources are considered a sub-component of a task in *i**, which is not distinguished from a sub-goal, sub-task, or sub-softgoal. In iStar 2.0, as resources are different from goals and task in nature, a different kind of link – "NeededBy" is suggested. Its implication is that resources are leave nodes, they will not be further refined, they are only checked for availability or not.

3.6 Removing some of the contribution links in *i**

In iStar 2.0, four types of contribution links are defined: help, hurt, make and break. Other contribution link types, such as: some+, some-, unknown, AND, OR are discarded. The implication of this change is that, contribution links are evaluated individually, we only consider weak or sufficient, positive or negative contributions from a source element to a quality attribute. This also improves the clarity of the models.

4 Discussions

In summary, iStar 2.0 clarifies ambiguities in *i** modelling framework, which makes the adoption by students and engineers learning the modelling syntax and semantics easy. For *i** users, it will not require much effort to understand the changes and adopt it in new cases and applying the changes in existing *i** models. Some minor points require further deliberation are as follows:

- Is a role allowed to participates-in an agent? While there is no logical explanation for it, in the iStar 2.0 meta-model, it is not prohibited?
- Is instantiation relation in actors allowed in iStar 2.0? Sometimes the modeller may want to express scenarios at an instance level?
- Is a quality attribute evaluated by itself or together with the element it qualifies? In other words, whether a quality attribute is a standalone element or is only meaningful together with an entity?
- A major problem yet to be addressed are using views of actors as a measure to control scalability of model.

In today's organization, social and strategic analysis is often supported with operational data as evidence for social dependencies and influential factors. Thus, automated elicitation of social modelling concepts and relations are considered an effective way to obtain social relation models as in *i**.

Acknowledgements

This paper is partially supported by the National Natural Science Foundation Project (Grant No. 61432020) and National Science and Technology Support Program (project no. 2015BAH14F02). Tong Li acknowledges the support of BJUT Startup Funding No.007000514116022.

References

1. E. Yu. Strategic Modelling for Enterprise Integration, Proceedings of the 14th World Congress of International Federation of Automatic Control (IFAC'99), July 5-9, 1999, Beijing, China. pp. 127-132. Permagon, Elsevier Science.
2. E. Yu and J. Mylopoulos. Modelling Organizational Issues for Enterprise Integration, Proceedings of International Conference on Enterprise Integration and Modelling Technology, October 28-30, 1997, Turin, Italy.
3. Agha G A. Actors: A model of concurrent computation in distributed systems[R]. MASSACHUSETTS INST OF TECH CAMBRIDGE ARTIFICIAL INTELLIGENCE LAB, 1985.
4. d'Inverno M, Kinny D, Luck M, et al. A formal specification of dMARS[C]//International Workshop on Agent Theories, Architectures, and Languages. Springer Berlin Heidelberg, 1997: 155-176.

5. Dalpiaz F, Franch X, Horkoff J. istar 2.0 language guide[J]. arXiv preprint arXiv:1605.07767, 2016.
6. Yu E, Liu L. Modelling trust for system design using the i* strategic actors framework[M]//Trust in Cyber-societies. Springer Berlin Heidelberg, 2001: 175-194.
7. Liu L, Yu E. Designing information systems in social context: a goal and scenario modelling approach[J]. Information systems, 2004, 29(2): 187-203.
8. Liu L, Liu Q, Chi C H, et al. Towards a service requirements modelling ontology based on agent knowledge and intentions[J]. International Journal of Agent-Oriented Software Engineering, 2008, 2(3): 324-349.
9. Sun J, Liu F, Zhang H, et al. Understanding the Diversity of Services Based on Users' Identities[C]//International Conference on Advanced Information Systems Engineering. Springer Berlin Heidelberg, 2011: 612-626.
10. Chung L, do Prado Leite J C S. On non-functional requirements in software engineering[M]//Conceptual modeling: Foundations and applications. Springer Berlin Heidelberg, 2009: 363-379.