

A Learner-Friendly Approach for Using the iStar Modeling Framework: an Ongoing Study

Yunduo Wang¹, Jinlian Du¹ and Tong Li¹

¹ Faculty of Information Technology, Beijing University of Technology, Beijing, 100124, China

Abstract

The utility and practicality of iStar framework can be affected by various factors, such as its conceptual model, ways of modeling, and supporting tools. In this paper, we report our ongoing empirical work that aims to investigate the experience of iStar beginners, based on which we explore how beginners can learn and use iStar framework more easily. In particular, we present a research design of this work and discuss initial results from a specific case. In the future, we plan to conduct a series of case studies, based on which we want to propose a practically efficient iStar modeling procedure and a corresponding CASE tool.

Keywords

Learner-friendly, iStar 2.0, Case study

1. Introduction

Requirements modeling is an effective way for requirements analysis, and i* modeling is one of the powerful means and gets widely recognized [1]. After more than a decade of development, there are many variants of i* for various analyses. The inconsistency of concepts is not conducive for beginners to learn and use i*. Recently, Dalpiaz et al. [2] have released iStar 2.0, which evolves basic concepts of i* into a consistent and clear set of core concepts. In this paper, all concepts we used are in line with iStar 2.0, and hereafter we use iStar instead of i* to keep consistency.

Last year we conducted a controlled experiment aimed to compare graphical and textual modeling in iStar framework [3]. As a result, we find that many factors can improve the usefulness and practicality of iStar framework. Specifically, the main factors come from three aspects, the conceptual model of iStar framework, modeling methods (e.g., graphical or textual modeling), and modeling tools. In this work, we mainly focus on investigating beginners' understanding of the conceptual model, which is important for promoting the practical application of iStar framework. We expect to understand the challenges it faces in practical applications so that solutions can be designed specifically, e.g., by providing appropriate tool support.

Abrahão et al. [4] conducted a series of experiments to compare iStar and another modeling language. The process by which they conducted their experiments gives us a good example.

Proceedings of the 14th International iStar Workshop, October 18-21, 2021, St. Johns (NL), Canada
EMAIL: itong@bjut.edu.cn (T. Li)



© 2021 Copyright for this paper by its authors.
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
CEUR Workshop Proceedings (CEUR-WS.org)

Ournani et al. [5] conducted a case study via interviews in an industrial context. We refer to their methods for our interview design. Ruiz et al. [6] reported on their employment of the proposed iStar 2.0 in a master-level course, including their research design, results, and experience. They focused on a complete process of using iStar 2.0, and we are similar to them here. They provided the existing iStar 2.0 artifacts to participants, and unlike them, our participants need to build their iStar models.

In this paper, we systematically design a case study protocol, with the aim of exploring beginners' experience with iStar 2.0. We then conduct a specific case, and report its results. We focus on investigating an easy way to learn and use iStar framework from a beginner's perspective. In the remaining parts of this paper, we introduce the case study protocol we designed in Section 2. Then we present the initial results from a specific case in Section 3 and discuss threats to validity in Section 4. Finally, we conclude and describe future work in Section 5.

2. Case study protocol

We plan to perform a case study to attain beginners' experience of learning to use iStar 2.0. For conducting the case study, we followed the guidelines of Wohlin et al. [7] to design the research protocol. We detail our protocol in the following subsections.

2.1. Research objectives

Improving the learning experience of beginners can effectively promote the widespread adoption of iStar. Thus, we want to find out how to make it easier for beginners to learn and use iStar 2.0. To this end, we formulate six research questions (RQs).

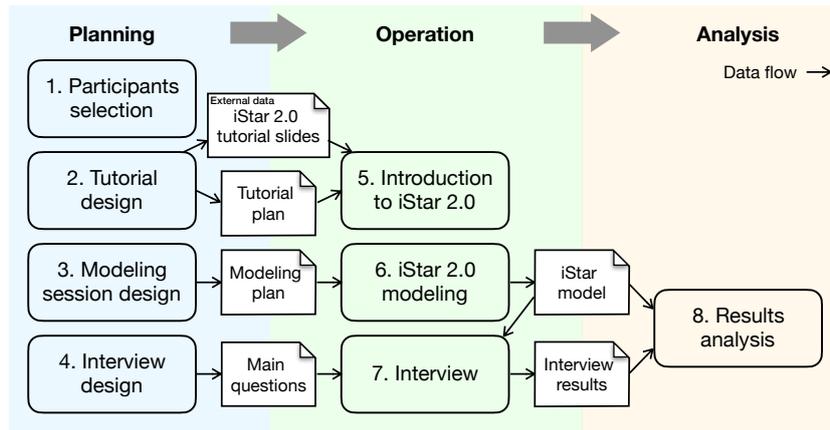
- RQ1: How difficult is it for beginners to learn iStar 2.0, and what are the reasons for this?
- RQ2: What is an effective way to teach beginners iStar 2.0?
- RQ3: What are the modeling processes adopted by beginners?
- RQ4: What are the difficulties of iStar modeling in practice?
- RQ5: Can these difficulties be mitigated or solved by a tool?
- RQ6: What requirements do beginners have for an iStar modeling tool?

2.2. Case study process

To answer the six RQs, we design our case study process as shown in Figure 1, which consists of eight steps, falling into three parts. In the current section, we will detail the design of the first two parts, Planning, and Operation in the following two subsections, i.e., Section 2.3 and Section 2.4. As for the third part, we report the initial results we got from a specific case in the next section.

2.3. Planning

In the planning part, we first selected the participants. Then we designed the iStar 2.0 tutorial, modeling and interview sessions for our participants.



We performed the eight steps of the three parts according to the serial number.

Figure 1: Case study process

2.3.1. Participants selection

The selection of participants determines the validity of the whole case study. Here we make a balance between operability and representativeness. We plan to select students at our university who have developed software with real users as participants. Although they are students, we believe their experience is similar to engineers new to the industry. These students' experience in practical software development is just the right thing to build a bridge between academia and industry. We plan to carry out our case study with such students one by one until we reach saturation status, i.e., the status we cannot gain any new information or viewpoint from new participants [8].

2.3.2. Tutorial design

We choose the iStar 2.0 Tutorial slides¹ as our handouts. The explanation is partial without involving quality and relations associated with it to simplify this procedure. We assign one of our authors to teach iStar 2.0 to participants. During the tutorial, we allow participants to ask questions at any time. We do not limit the length of the tutorial. We will not end this session until participants have no more questions about iStar 2.0.

2.3.3. Modeling session design

For each participant, the modeling scenario is the actual application scenario of the software the participant has developed. If a participant has developed more than one software in practical use, we will choose the largest one. For modeling, we plan to use a whiteboard way. We believe that the whiteboard modeling is a pure approach and will allow us to exclusively focus on the conceptual model of iStar framework rather than various usability issues of modeling tools. Specifically, we provide several sheets of white A4 paper, a black pen, and a red pen to

¹<https://www.dropbox.com/s/4l2k4tbywb8wekk/iStar-tutorial-online.pdf?dl=0>

participants for modeling. Here we choose black and red pens as they are the two pen colors we use most often in our daily lives. During the modeling process, participants could refer to the tutorial slides and ask questions at any time. Similar to the tutorial, we still have no time limit here. The modeling session will not end until participants think they have finished modeling. In addition, as a final result of the modeling session, we ask each participant to provide an iStar model that we can fully understand.

2.3.4. Interview design

Referring to the work of Ournani et al. [5], we believe that semi-structured interviews would be similarly helpful in our case. The pre-defined main questions allow the interview to focus on our research questions and objectives, while we can ask any follow-up questions based on the participants' answers to obtain more details.

We expect the views expressed by the participants in our interview could be able to answer the RQs, so we set the main questions in the interview according to our RQs. Figure 2 shows these questions and how they relate to the RQs and our research objectives.

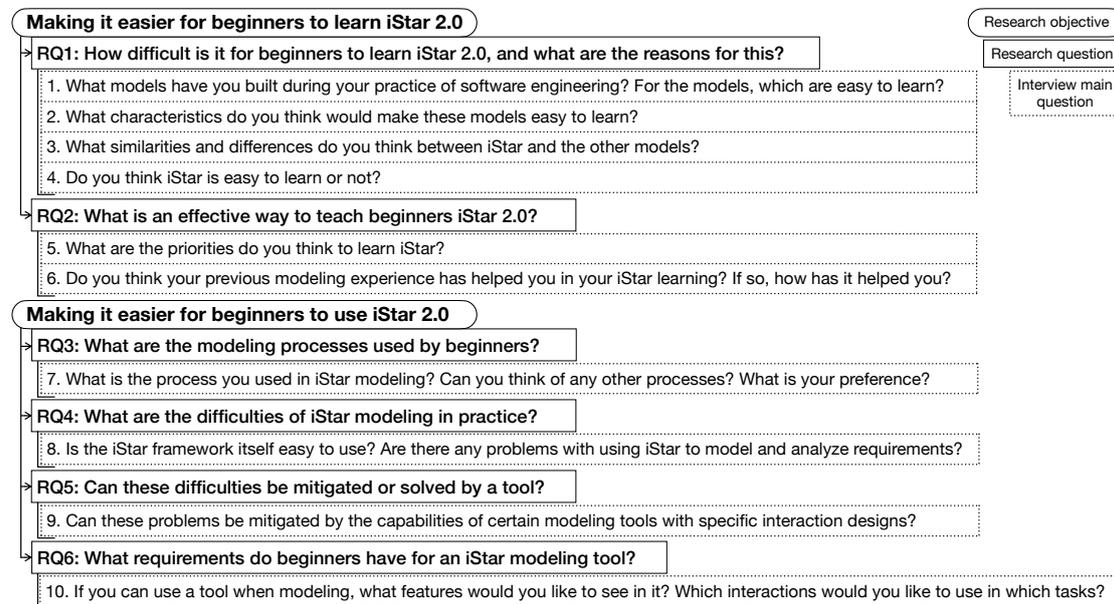


Figure 2: Relationships among research objectives, research questions, and interview main questions

2.4. Operation

One of our authors experimented with one participant. The author recorded the whole process with the consent of the participant. The author first introduced iStar 2.0 to the participant and then accompanied the participant in modeling. Finally, the author interviewed the participant. In the end, it took 20 minutes to the tutorial, 50 minutes to modeling, and an hour to interview.

In addition to the audio recordings, the author took note of what the participants said during the interview. It is worth noting that what the author took down had been confirmed with the participant to ensure that these notes were what the participant wanted to express.

3. Initial results from a specific case

We report the interview results from a specific case in this section.

3.1. Background information for the interview

For privacy and confidentiality purposes, we use a code name P1 as the participant's name. P1 is a fourth-year undergraduate at Beijing University of Technology, majoring in computer science. In the past three years, P1 has developed and operated a modeling platform. To date, the platform has been running for two years and has over 200 unique real users. P1 acknowledges the importance of requirements modeling and has some experiences with use case modeling, but has no experience with iStar modeling.

For the modeling session of our experiment, we asked P1 to use iStar to model and analyze the requirements in the scenario based on real situations where users are currently using P1's modeling platform. The final iStar model P1 built has 4 actors, 20 intentional elements, 21 intentional element links, and 8 dependencies, 53 model elements in total.

3.2. Interview results

According to our interview design in Section 2.3.4, P1's responses in the interview should answer the RQs. Therefore, we report the interview results in the order of the RQs as following.

RQ1: How difficult is it for beginners to learn iStar 2.0, and what are the reasons for this?

P1 believes that of all models P1 has built, the UML class diagrams, Entity-relationship diagrams, Data flow diagrams, Use case diagrams, and iStar 2.0 are in ascending order of difficulty to learn. For P1, UML class diagrams are easy to learn since they are very similar to code structures. P1's experience in programming exactly can help P1 learn UML class diagrams. P1 explained that requirements modeling is usually more difficult than design modeling, and the complexity of the iStar modeling framework makes it even more difficult than use case diagrams.

RQ2: What is an effective way to teach beginners iStar 2.0?

P1 believes that it is important and difficult to extract requirements from users' perspectives when first starting to model. If we want to teach beginners iStar 2.0 effectively, we should focus on it. In addition, P1 said that the similarities between the learned model and iStar could generate help in learning iStar.

RQ3: What are the modeling processes used by beginners?

Regarding the modeling process, P1 stated that P1 would list only the user requirements at the beginning, and then the user requirements would elicit the system requirements. In addition, P1 said that it was also a way to list a user requirement and then immediately analyze its corresponding system requirements for modeling. However, P1 believes that such a process can easily lead to confusion.

RQ4: What are the difficulties of iStar modeling in practice? **RQ5:** Can these difficulties be mitigated or solved by a tool? **RQ6:** What requirements do beginners have for an iStar modeling tool?

P1 thinks that iStar modeling is tough to get started, but the modeling results are helpful. P1 believes that the use of tools can mitigate some of the difficulties. For instance, an intelligent modeling tool that can extract model elements from requirements text automatically.

4. Threats to validity

In this section, we discuss threats that might affect the validity of our research result. Here we adopt a classification scheme from Runeson et al. [9] that focuses on case study research in software engineering.

Construct validity In our interviews, misunderstandings and misinterpretations of the iStar framework or question descriptions can affect construct validity. To mitigate these threats, we encourage participants to think aloud so that we can correct them if necessary. Another aspect is whether the difficulties originate from the iStar metamodel or are rather inherent to goal modeling and goal orientation. As the start of a long-term study, we plan to make the distinction after we have figured out the difficulties of the iStar framework as comprehensively as possible. In other words, this is part of our future work.

Internal validity Our experiment requires participants to have software development experience, so the selection of participants is not random, which could pose a threat to internal validity. The quality of the requirements involved in the modeling session may also affect the results of the experiment. Ambiguous requirements descriptions may also make iStar modeling difficult.

External validity Although our participants are only students, their real software development experience can mitigate the threat to external validity to some extent. However, the external validity still needs to be further demonstrated through experimentation in industry settings.

Conclusion validity The small sample size is a general threat to conclusion validity. We will mitigate this threat by conducting experiments with much more participants in our future work.

5. Conclusions and future work

In this paper, we present the design of a case study that we plan to conduct to explore the experiences of iStar beginners. We intend to find an easier way for beginners to learn and use iStar framework based on the results from the case study. Specifically, we have carefully designed a research protocol and conducted the study through a specific case. We report our initial results from that case.

As for future work, firstly, there is no doubt that we will continue the research with more participants. In addition, based on the experience we gained in this experiment with the specific case, we need to ensure the quality of the requirements for modeling before our case study. Secondly, we will also analyze the data from the modeling session. Specifically, these data

include the mistakes made by participants, the time they spent on each procedure throughout the modeling process, and the questions they asked during the modeling session. Lastly, we will also analyze the content of the modeling results from participants, including how they use the red and black pens during their modeling. Based on the results, we plan to propose an iStar modeling procedure and a corresponding CASE tool. We expect these will be effective in practice.

Acknowledgments

This work is partially supported by the National Natural Science of Foundation of China (No.61902010), and the Project of Beijing Municipal Education Commission (No.KM202110005025), and the International Research Cooperation Talent Introduction and Cultivation Project of Beijing University of Technology (No.2021C01).

References

- [1] J. Horkoff, F. B. Aydemir, E. Cardoso, T. Li, A. Maté, E. Paja, M. Salnitri, L. Piras, J. Mylopoulos, P. Giorgini, Goal-oriented requirements engineering: an extended systematic mapping study, *Requirements Engineering* 24 (2019) 133–160.
- [2] F. Dalpiaz, X. Franch, J. Horkoff, *istar 2.0 language guide*, arXiv:1605.07767 (2016).
- [3] W. Liu, Y. Wang, Q. Zhou, T. Li, Graphical modeling vs. textual modeling: An experimental comparison based on istar models, in: *2021 IEEE 45th Annual Computers, Software, and Applications Conference (COMPSAC)*, 2021.
- [4] S. Abrahão, E. Insfran, F. G.-L. de Guevara, M. Fernández-Diego, C. Cano-Genoves, R. Pereira de Oliveira, Assessing the effectiveness of goal-oriented modeling languages: A family of experiments, *Information and Software Technology* 116 (2019) 106–171.
- [5] Z. Ournani, R. Rouvoy, P. Rust, J. Penhoat, On reducing the energy consumption of software: From hurdles to requirements, in: *Proceedings of the 14th ACM / IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM), ESEM '20*, Association for Computing Machinery, New York, NY, USA, 2020.
- [6] M. Ruiz, F. B. Aydemir, F. Dalpiaz, Using conceptual models in research methods courses: An experience using istar 2.0, in: *CEUR Workshop Proceedings of the 2nd International iStar Teaching Workshop co-located with ER 2017*, volume 1954, 2017, pp. 48–57.
- [7] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, A. Wesslén, *Experimentation in software engineering*, Springer Science & Business Media, 2012.
- [8] J. Corbin, A. Strauss, *Basics of Qualitative Research*, 3rd edn, SAGE, Los Angeles, 2008.
- [9] P. Runeson, M. Höst, Guidelines for conducting and reporting case study research in software engineering, *Empirical Software Engineering* 14 (2008) 131.