Information and Software Technology 187 (2025) 107839

Contents lists available at ScienceDirect

Information and Software Technology

journal homepage: www.elsevier.com/locate/infsof

Understanding the challenges and requirements for facilitating iStar learning: An empirical study with iStar learners☆

Tong Li [a,*], Qixiang Zhou [a], Yunduo Wang [b], Haonan Xiong [a], Wenxing Liu [a], Ning Ge [b]

[a] School of Computer Science, Beijing University of Technology, No. 100 Pingleyuan, Beijing, 100124, China
[b] School of Software, Beihang University, No. 37 Xueyuan Road, Beijing, 100191, China

ARTICLE INFO

Keywords:
Empirical study
iStar learning
Modeling education
Bloom's taxonomy
Grounded theory

ABSTRACT

Context: As research into the iStar framework continues, more researchers are recognizing its important role in requirements analysis. However, the steep learning curve of the iStar framework hinders widespread adoption in practice. An increasing number of recent studies focus on the practical application of the iStar framework, revealing difficulties in learning and practicing iStar.
Objective: This paper aims to investigate and provide a thorough understanding of the challenges faced in learning and practicing the iStar modeling framework, utilizing empirical research and grounded theory.
Method: Accordingly, we systematically designed and conducted an empirical study that involved ten iStar learners and three iStar lecturers to discover the difficulties of learning iStar. Utilizing the revised Bloom's taxonomy, we formulated our research questions and, based on the study results, developed a systematic theory about these difficulties. We then elicited a series of requirements to facilitate iStar learning employing Strauss and Corbin's coding paradigm from grounded theory. Furthermore, we implemented and deployed an online prototype tool inspired by this study to provide (semi-)automatic support for iStar learning and practicing.
Results: The study identified key difficulties both learners and lecturers face when engaging with the iStar framework. Through grounded theory, we developed a systematic understanding of these challenges and formulated requirements to address them. These requirements were implemented into a prototype tool, aiming to facilitate iStar learning and practice.
Conclusions: This paper presents an empirical study involving both beginners and lecturers in iStar modeling, providing valuable insights into the difficulties faced in learning and teaching the iStar framework. By addressing these challenges through derived strategies, proposed requirements, and the developed tool, we aim to ease the learning curve of iStar, thereby promoting its wider adoption in both educational and practical contexts.

1. Introduction

Goal-oriented requirements engineering (GORE) has drawn much attention from academia and industry, which is pivotal in capturing stakeholders' objectives and aligning system functionalities with requirements and goals. The iStar modeling framework (henceforth, we use iStar for short) is a well-explored GORE approach [1,2], which has been investigated for decades [3]. However, it is challenging to deal with flexible real-world projects due to the steep learning curve [4–8], making it difficult to learn and practice, hence preventing iStar modeling from widespread use.

In the modeling education field, researchers have made diverse efforts to promote the education and practice of iStar, sharing different reports on experiences in teaching and promoting the use of iStar. Wang et al. [9] investigate the experiences of iStar learners, aiming to make iStar easier for learners to learn and use. Gonçalves et al. [10] present the usage of iStar modeling in an industrial project, surveying opinions of using iStar modeling from participants. Moreover, Ruiz et al. [11], and Dalpiaz [12] report their experience in teaching iStar. Although these efforts contribute valuable insights, they primarily focus on implementation experiences rather than systematically identifying the underlying learning challenges. This gap indicates a need for a

☆ This work was supported by the Central Guidance for Local Scientific and Technological Development Fund (No. 2024ZY0124) and the National Natural Science Foundation of China (No. 62162051).
* Corresponding author.
E-mail addresses: litong@bjut.edu.cn (T. Li), zhouqx@emails.bjut.edu.cn (Q. Zhou), wangyunduo@buaa.edu.cn (Y. Wang), xiong_haonan@emails.bjut.edu.cn (H. Xiong), liuwenxing@emails.bjut.edu.cn (W. Liu), gening@buaa.edu.cn (N. Ge).

https://doi.org/10.1016/j.infsof.2025.107839
Received 7 January 2025; Received in revised form 3 July 2025; Accepted 15 July 2025
Available online 5 August 2025

comprehensive investigation into the difficulties learners face when adopting iStar.

To address this gap, we advocate for a systematic investigation of the challenges and requirements associated with learning complex modeling methods like iStar. Our objective is to uncover the specific difficulties learners encounter and to develop strategies to facilitate more effective learning and practice. In this paper, we present an empirical study designed to achieve these goals by involving both iStar learners and lecturers. In this paper, we take iStar as a typical example of such cases and systematically design and conduct an empirical study to understand the difficulties learners face in learning and practicing iStar. Our participants included both learners and lecturers in iStar modeling. Our study utilizes revised Bloom's taxonomy to design research questions and analyze results using grounded theory to identify factors of difficulties faced by learners, including a set of strategies as requirements for effective iStar learning and practice. Our work paves the way for promoting modeling education. For one thing, parts of the study results are also applicable to other modeling methods; for another, the research procedure can be replicated to investigate other modeling methods. Specifically, the contributions to this paper are as follows:

1. We systematically design and conduct an empirical study utilizing Bloom's taxonomy with ten iStar learners to understand their difficulties in learning and practicing iStar. In addition, the perspectives of the three lecturers, obtained through their observations during the tutorial session from the case study, were enlisted as supplemental insights to the beginners' viewpoint.
2. Through Strauss and Corbin's grounded theory methodology to analyze the empirical study results, we develop a systematic theory of these challenges and derive detailed requirements to facilitate iStar learning.
3. A prototype tool is developed, which contributes to the satisfaction of the elicited modeling requirements and pragmatically facilitates iStar learning and practicing.

In the remaining sections of this paper, we present the related work about promoting iStar learning and practicing iStar in Section 3. The protocol and results of the empirical study are depicted in Section 4, analyzed, and elicited a set of requirements using grounded theory in Section 5. According to these requirements, we implement and deploy a prototype tool that implements them in Section 6. Finally, we conclude this paper in Section 7.

2. Background

In this section, we provide the background of our proposal, including the iStar modeling framework and Bloom's taxonomy used in this study.

2.1. iStar modeling framework

The iStar modeling framework is a goal- and agent-oriented modeling and reasoning framework primarily employed in the early stages of requirements engineering [13]. Initially introduced in the mid-1990s, it quickly gained recognition within the research community due to its flexibility and practical utility. The current standardized iteration, iStar 2.0, offers a refined and consistent set of concepts and notations [1]. This paper adheres to the iStar 2.0 standard throughout.

The iStar notation consists of various elements that represent actors and their goals, as illustrated in Fig. 1. Specifically, actors pursue goals and can be classified as roles or agents. Relationships among actors are represented either by actor association links or by social dependencies, which denote dependencies between actors and intentional elements. In a dependency relationship, one actor named "depender" relies on another actor named "dependee" to provide an intentional element named the "dependum" necessary to achieve a certain goal. Actor association links, including "is-a" and "participates-in", are binary relationships connecting one actor directly to another.

Intentional elements define actors' desires or objectives, encompassing goals, qualities, tasks, and resources. Goals explicitly represent actors' desired states and can be hierarchically decomposed using intentional element links. Notably, refinement links form an n-ary relationship structure, breaking down goals into sub-tasks through logical operators such as "AND" and "OR". Additional intentional element relationships, such as "needed by", "contribution", and "qualification", further specify interactions among these elements.

iStar models consist of several distinct views designed to capture varying levels of detail: the Strategic Dependency (SD) view, the Strategic Rationale (SR) view, and the Hybrid SD/SR view, which integrates aspects of both SD and SR views. The SD view emphasizes actors and their interactions, particularly actor association links such as "is-a" and "participates-in" and social dependencies. In contrast, the SR view expands upon this by incorporating internal intentional elements and their relationships with individual actors. The Hybrid SD/SR view balances detail and clarity by selectively displaying the strategic rationale for certain actors, thus providing a comprehensive yet manageable representation of complex systems [1].

2.2. Bloom's taxonomy

In order to systematically investigate the learning challenges, we adopt Bloom's revised taxonomy [14], a widely recognized framework for categorizing educational objectives based on cognitive complexity. This taxonomy provides a structured approach to analyze learners' cognitive processes, which is particularly relevant for understanding the multifaceted difficulties in mastering iStar's complex concepts and practices. Bloom's taxonomy is a classification of learning outcomes that categorizes educational objectives into different levels of complexity and specificity, which is widely used to design learning outcomes, assess student performances, and guide instructional strategies [15].

Bloom's taxonomy encompasses three different domains of learning: cognitive, affective, and psychomotor. Most research focuses on the cognitive domain, which is related to the way people recall knowledge, comprehend, and critically think. Within the cognitive domain, the revised taxonomy identifies six levels of cognitive processes, arranged in increasing order of complexity [16]:

1. **Remember**: Remember is the foundational level of the revised Bloom's taxonomy and involves the retrieval, recall, or recognition of relevant knowledge from long-term memory. At this level, students are expected to memorize and recall specific facts, principles, or procedures without necessarily understanding their deeper implications or connections. Corresponding assessments typically include tasks that require students to retrieve or recognize information.
2. **Understand**: Understand involves demonstrating comprehension through one or more forms of explanation. At this level, students move beyond simple recall and memorization to show that they can grasp the meaning and significance of the learned material. Learners should be able to interpret, classify, compare, contrast, explain, and summarize information in their own words. Assessments targeting this level may include tasks that require students to explain concepts, provide examples, or compare and contrast ideas.
3. **Apply**: Apply involves using the information or skill in a new situation. At this level, students move beyond understanding to actively apply their knowledge to solve problems in novel contexts. Learners should be able to take the concepts, principles, or procedures they have learned and apply them to real-world situations or hypothetical scenarios. Assessments targeting this level may include tasks that require students to apply knowledge to solve problems, complete projects, or analyze case studies.

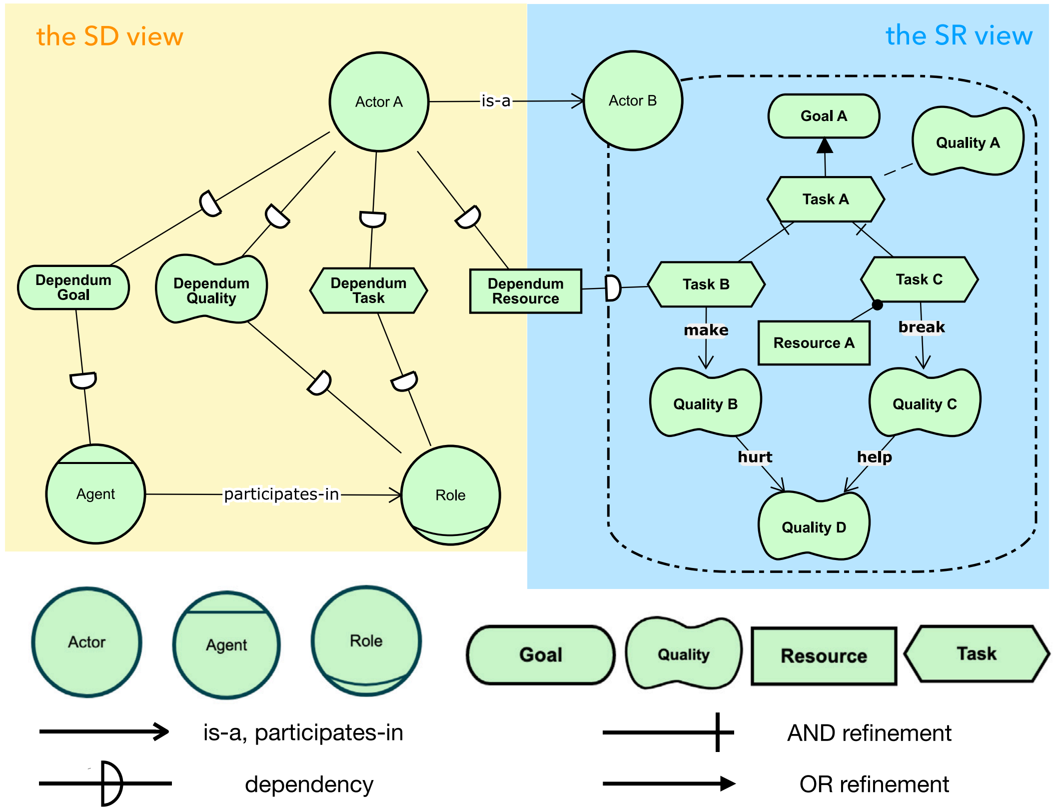

Fig. 1. The view of an iStar model.

4. **Analyze**: Analyze involves breaking material into its constituent parts and determining how the parts relate to one another and/or to an overall structure or purpose. At this level, students move beyond applying knowledge to analyzing and deconstructing information, ideas, or systems to better understand their components, relationships, and organizational principles. Learners should be able to examine the material, identify its key elements, and understand how these elements interact or contribute to the whole. Corresponding assessments may include tasks that require students to analyze case studies, compare and contrast ideas, or examine the structure and components of a system.
5. **Evaluate**: Evaluate involves making judgments based on criteria and standards. At this level, students move beyond analyzing information to assessing the value, quality, or effectiveness of ideas, products, or processes based on established criteria or standards. Learners should be able to evaluate the merits and drawbacks of different approaches, solutions, or arguments and provide reasoned justifications for their judgments. Assessments targeting this level may include tasks that require students to critique an argument, assess the effectiveness of a solution, or evaluate the quality of a product based on established criteria.
6. **Create**: Create is the highest level of the revised Bloom's taxonomy, involving putting elements together to form a new coherent or functional whole or reorganizing elements into a new pattern or structure. At this level, students move beyond evaluating ideas or products to generating, planning, or producing original work. Learners should be able to synthesize knowledge and skills from various domains to create new and innovative solutions, ideas, or products. Corresponding assessments may include tasks that require students to develop original solutions to complex problems, design innovative products or systems, or create new knowledge through research or artistic expression.

By applying Bloom's taxonomy, we can categorize and examine the challenges at different cognitive levels that learners face when engaging with iStar, such as understanding, application, and analysis. This approach enables us to tailor strategies to specific cognitive stages, thereby enhancing the effectiveness of iStar education.

3. Related work

In this section, we review empirical studies that promote goal-modeling education and related works that utilize the taxonomy of software engineering education.

3.1. Empirical studies for goal modeling

Researchers have reported experience in educating or promoting goal modeling from their applications. Maslov et al. [17] present a comprehensive bibliometric review of research on conceptual modeling education across various disciplines, including information systems, business engineering, and natural sciences. By analyzing over a thousand publications, they provide insights into prevalent conceptual modeling subtypes, research methodologies, and associated learning theories, uncovering significant conceptual, theoretical, and methodological gaps within the field.

Wang et al. [9] present an ongoing empirical study investigating the experiences of iStar learners, with the goal of making iStar easier for learners to learn and use. They systematically design a case study protocol focused on learners' experiences with iStar 2.0. The participants in this study need to build their own iStar models. The authors report initial results from a specific case and discuss threats to validity. Their future plans include conducting more case studies to propose an efficient iStar modeling procedure and a corresponding CASE tool to aid learners.

Gonçalves et al. [10] use iStar modeling and analysis requirements in an industrial project to evolve an e-commerce application. After the collaborative modeling, the participants are surveyed to identify their opinions of using iStar modeling. Ruiz et al. [11] report on their use of iStar 2.0 in the master-level course Advanced Research Methods at Utrecht University. During this course, students conduct a design science project where iStar 2.0 artifacts are evaluated in an experimental setting. The authors present the course, explain how they employ iStar 2.0, and discuss their teaching experience including lessons learned. They contribute by promoting the use of conceptual modeling in teaching environments, reporting on the design of comparative experiments using iStar 2.0 artifacts, applying the design science method for evaluating iStar 2.0 artifacts in classroom settings, and discussing the use of iStar 2.0 as teaching material for research methods courses in information systems programs. Dalpiaz [12] reports on his experience of teaching iStar and its basic, essential dialect called simple iStar to over 130 first-year students of a bachelor's degree in information science. He presents the intended learning outcomes and activities, introduces the simple iStar dialect used in the labs, and discusses how the gained knowledge was tested in the final exam.

Nkamaura and Tachikawa [18] propose a new method for requirements engineering education that combines group-work role-play training to elicit customers' real requirements and a software agent that plays the role of a domain expert. The experimental assessment with undergraduate students shows that students' requirements modeling skill levels improve, but the role-play scenarios need to be carefully designed to provide all necessary information in the problem domain. Gralha et al. [19] research whether gender differences matter when interacting with iStar models to promote iStar and similar goal modeling frameworks. They conducted a quasi-experiment and characterized 180 participants based on various gender-related facets. These findings could be utilized by leveraging people's diversity to obtain benefits in iStar learning. Zhu et al. [20] propose a comprehensive framework for systematically recognizing and digitizing iStar hand drafts via customized object detection. Besides, this paper establishes a dataset of 630 hand drafts for further evaluation. The evaluation results show the practical applicability of this approach.

Singh et al. [21] propose an extension to iStar for modeling human-centric aspects of end-users, such as gender, age, emotions, personality, and cultural factors. They emphasize the importance of capturing these characteristics during early software modeling to prevent usability issues and user frustration. The authors integrate persona and contextual modeling into iStar to represent diverse user attributes, illustrating their approach with examples from a flight booking system and a smart home system for the elderly. Their user evaluation studies indicate that the extended iStar language facilitates better understanding and consideration of human-centric requirements, demonstrating the necessity for further research in human-centered modeling. Ruiz and Franch [22] investigate the pedagogical benefits of using iStar 2.0 goal models for teaching data structures to undergraduate computer science students. Through a quasi-experiment with 54 third-semester students during Fall 2022, they compare a control group using traditional teaching methods against an experimental group exposed to data structures described with iStar 2.0. Their assessment reveals that while students exposed to goal models demonstrate higher validity in describing data structures, they require more time to complete tasks. Abdoul Soukour et al. [23] propose an innovative approach for generating KAOS goal models by leveraging knowledge graphs and natural language processing techniques. Rather than automating the entire goal model creation process, they offer interactive assistance to designers for gradually refining their goals. Their method utilizes semantic similarity, natural language inference, sentiment analysis, and graph-to-text generation to extract relevant knowledge from domain knowledge graphs. The approach is demonstrated by applying it to a flood management system, using a handcrafted domain knowledge graph. Their evaluation shows that 35.2% of triples from the KG were highlighted, with 73.7% of these being useful for goal refinement. Batista et al. [24] conducted an empirical study to evaluate the effectiveness of Goal-Oriented Requirements Engineering (GORE), specifically the KAOS methodology, in teaching requirements elicitation for AI-based systems. They engaged 34 undergraduate software engineering students through lectures, training sessions, and practical exercises where students applied KAOS to elicit requirements for an AI system. Data was collected through requirement analysis and surveys. Results showed 88% of students applied KAOS correctly or with minor inadequacies, indicating good learnability. Students reported benefits including GORE's structured approach and goal decomposition capabilities, while challenges included determining refinement stopping criteria and managing diagram complexity.

While these studies provide valuable insights into educational practices, they often fail to systematically analyze the specific challenges learners encounter during the learning process. This limitation highlights the importance of examining the challenges faced by iStar learners and analyzing the difficulties within the iStar learning process.

3.2. Conceptual modeling studies using Bloom's taxonomy

The application of Bloom's taxonomy in conceptual modeling education has garnered increasing attention in recent years. Bork [25] proposes a framework for comprehensively teaching conceptual modeling and metamodeling based on Bloom's revised taxonomy. The framework is used to evaluate a Smart City teaching case, revealing gaps in the coverage of knowledge dimensions. Bork discusses the need for incorporating evaluation and metacognitive dimensions in future iterations of the case.

Bogdanova and Snoeck [26] conduct a study to classify learning outcomes in domain modeling education using Bloom's revised taxonomy. They analyze 291 exercises and tasks from various sources, including books, online courses, and university exams. The results show an uneven distribution of cognitive processes and knowledge dimensions, with gaps in scaffolding and assessment. Their work highlights the need for a more comprehensive and systematic approach to teaching domain modeling.

Bogdanova and Snoeck [27] propose the CaMeLOT framework for teaching conceptual data modeling (CDM), adapting Bloom's taxonomy to better support CDM learning outcomes. They emphasize the challenges educators face in designing comprehensive curricula for CDM due to the ill-structured and context-dependent nature of data modeling problems. Their framework not only identifies various content areas and the necessary scaffolding but also offers 17 example learning outcomes that illustrate different levels of cognitive engagement and knowledge dimensions. This work aims to help educators streamline course design and improve assessment quality by offering a systematic and adaptable educational tool for conceptual modeling.

4. Research protocol

In this section, we propose a series of research questions and design the study process for facilitating iStar learning. We follow the methodology proposed by Wohlin et al. [28] to design our research protocol. Specifically, our empirical study meets the criteria by examining the challenges in learning and practicing iStar, which is a clearly identifiable phenomenon. Besides, this empirical study is conducted within an authentic educational context and employs multiple methods of data collection, including literature review, modeling artifact analysis, and semi-structured interviews.

To understand the challenges faced by beginners in learning iStar modeling, we involved iStar novices in our empirical study. Additionally, we invited experienced iStar lecturers to guide the beginners through the iStar modeling framework, covering key concepts, syntax, and practical usage with examples. These lecturers, with their expertise in teaching and practicing iStar, were able to provide professional analysis and validation of the beginners' perspectives.

The empirical study process is outlined in Fig. 2. Initially, we selected both the beginner participants and the lecturers. Following the selection, the beginners participated in an iStar tutorial led by the lecturers. Subsequently, the beginners engaged in iStar modeling exercises to deepen their understanding of the framework.

In parallel, we conducted a literature review on iStar challenges and tools, which provided additional insights to supplement our theoretical framework. Finally, we carried out interviews with the beginners, collecting and analyzing the results of these activities to inform our findings. Moreover, we perform a literature review on iStar challenges and iStar tools as supplemental insights to our theory. Lastly, we conduct interviews for beginners and collect and analyze the results from these operations.

4.1. Research questions

We aim to explore ways to make iStar easier for beginners to learn and practice. We believe that the challenges in iStar learning and practicing are closely related. On one hand, beginners often find it difficult to learn iStar due to its inherent complexity. To analyze these difficulties, we use the revised Bloom's taxonomy [14] and focus on the cognitive levels of remembering, understanding, and applying to design our research questions. These three levels are chosen because they are fundamental for beginners to master iStar modeling.

However, the lack of effective methods and tools to support iStar practice can also hinder the learning process. Therefore, we also aim to gain insights from iStar practice and propose targeted improvements that make iStar more accessible and effective for beginners. Thus, we design research questions as follows:

- **RQ1:** What specific challenges do beginners face when learning iStar at different cognitive levels, particularly understanding, applying, and analyzing?

In this study, we focus on the cognitive levels of understanding, applying, and analyzing from the revised Bloom's taxonomy [14]. Specifically, we conduct our research based on the cognitive levels defined in the CaMeLOT teaching framework [27] proposed for conceptual modeling. This work categorizes cognitive levels as shown in Table 1.

We carefully considered the rationale for selecting cognitive levels. For beginners learning a complex modeling method like iStar, our rationale for specifically choosing these three mid-tier levels is based on several practical considerations. First, mastering conceptual modeling frameworks inherently requires learners to go beyond simplistic knowledge recall. While the foundational level (remembering) is undoubtedly important, we found that the primary hurdles in comprehending iStar arise not from recalling definitions or terminologies but rather from effectively grasping their nuanced meanings (understanding), applying concepts correctly in practical modeling contexts (applying), and critically examining how different modeling constructs interrelate (analyzing). Indeed, tasks related to these three levels represent critical turning points determining whether learners gain meaningful competence in iStar modeling. Besides, the higher cognitive processes of evaluating and creating, though valuable in professional modeling contexts, require a level of expertise and practical experience beyond that typically possessed by novice learners. Beginners usually lack the necessary background to assess alternative modeling strategies (evaluating) critically or to generate innovative modeling approaches (creating). Introducing such advanced cognitive tasks prematurely could overwhelm beginners and hinder effective initial learning outcomes. Thus, we select understanding, applying, and analyzing levels in this study for beginners to perform the empirical study.

Understanding: At the understanding level, our research aims to investigate learners' cognitive processes involved in comprehending the classification of iStar elements, their relationships, graphical notation, conceptual models, and specific syntax rules. We focus on how learners exemplify various modeling terms, link terms with corresponding iStar notation, explain abstract modeling concepts using their own language. This research further identifies critical challenges learners encounter while performing these specific comprehension tasks and proposes strategies to enhance their understanding.

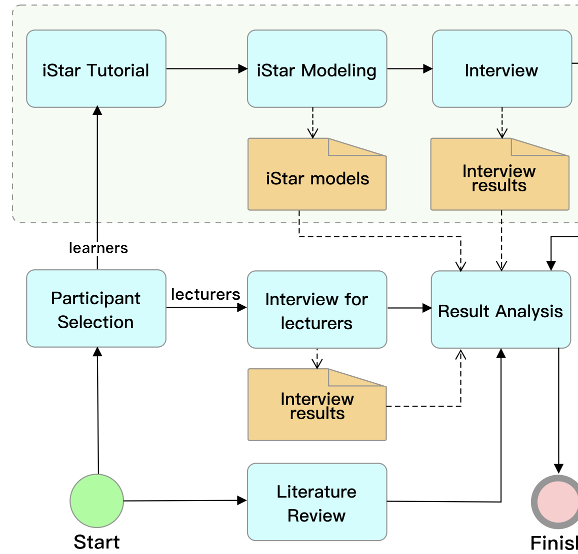

Fig. 2. Empirical study process.

Applying: At the applying level, this research concentrates on how learners utilize previously acquired iStar methods or modeling patterns to solve practical modeling tasks, including employing existing iStar modeling guidelines and heuristics to construct effective iStar models. Specifically, we examine learners' challenges in transforming real-world requirements into concrete iStar models and difficulties encountered during demonstrations and practical applications of the iStar modeling method. Our goal is to identify best practices that effectively support learners in applying iStar to real-world scenarios.

Analyzing: At the analyzing level, we explore how learners perform comparative analysis of elements and attributes within iStar models, including determining the completeness of model elements or attributes, distinguishing relevant from irrelevant information, and explicitly explaining their modeling choices. Furthermore, our research addresses how learners compare and contrast iStar with other modeling frameworks to reveal critical difficulties encountered during deeper analysis and comprehension of modeling characteristics and patterns. Based on this investigation, we propose targeted strategies to support learners in analytical tasks.

In addition, we conducted a literature review analyzing issues identified in other empirical studies involving iStar learners to further strengthen the completeness of our theory.

- **RQ2:** How can the iStar modeling process be improved to address the challenges in learning and practicing?

In this research question, we explore the challenges not only related to cognitive-level learning processes but also the limitations of the iStar modeling process. These challenges can potentially be overcome by enhancing the iStar modeling process and its associated tools and resources. By identifying and addressing these limitations, we aim to improve the overall experience of learning and applying iStar in practice.

Regarding the literature review, we synthesized improvement suggestions from other empirical studies on iStar learners to further consolidate the theoretical completeness.

4.2. Protocol in detail

In this section, we provide a detailed explanation of the steps outlined in Fig. 2, including participant selection, the iStar tutorial, the iStar modeling, the interview, and the result analysis session.

Table 1
Cognitive levels and associated tasks in conceptual modeling.

Cognitive Level	Typical Tasks
Remember	• Matching terms with definitions or recognizing the terms. • Giving definitions of terms. • Drawing graphical notation elements corresponding to terms. • Duplicating a model, pattern, or notation element. • Listing types of certain concepts or terms (e.g., associations).
Understand	• Exemplification — giving examples representing learned modeling terms. • Linking concepts with corresponding modeling notation elements. • Explaining modeling concepts in own words. • Translation tasks — converting notation to text. • Summarizing tasks (e.g., summarizing a requirements document or case). • Discussion tasks.
Apply	• Using a given pattern to solve a modeling task. • Applying previously learned heuristics or guidelines in practice. • Translating a model from one notation into another according to guidelines. • Demonstrating the use of a particular modeling method.
Analyze	• Ordering and comparison tasks. • Determining completeness or incompleteness of statements, elements, or attributes. • Distinguishing relevant from irrelevant information. • Explaining modeling choices (e.g., modeling something as a class or an attribute). • Determining which modeling pattern is used in a given model. • Mapping an analysis pattern onto a given problem.
Evaluate	• Checking correctness or finding mistakes in the used notation. • Identifying violations of conventions or rules. • Critiquing a solution or the choice of an example. • Identifying the most suitable analysis and design pattern for a modeling problem.
Create	• Developing solutions for given modeling tasks. • Generating classes, associations, or groups of classes based on requirements. • Designing model elements according to specifications. • Defining modeling patterns based on a series of problems and solutions.

Table 2
Main questions for learners corresponding to research questions.

Research question	Main questions for learners
(RQ1) What specific challenges do beginners face when learning iStar at different cognitive levels, particularly understanding, applying, and analyzing?	(MQ1) What are the challenges in remembering and understanding the concepts of iStar? (MQ2) What are the challenges in applying iStar, especially in iStar modeling? (MQ3) What are the challenges in analyzing iStar? Do you know the strengths and limitations of iStar compared with other modeling frameworks? (MQ4) Are there any other challenges in learning iStar?
(RQ2) How can the iStar modeling process be improved to address the challenges in learning and practicing?	(MQ5) What are the challenges and limitations of practicing iStar? (MQ6) Can these challenges mitigated by a tool? (MQ7) What requirements do you have to mitigate these challenges?

Table 3
Main questions for lecturers corresponding to research questions.

Research question	Main questions for lecturers
(RQ1) What specific challenges do beginners face when learning iStar at different cognitive levels, particularly understanding, applying, and analyzing?	(MQ1) What are the challenges in remembering and understanding the concepts of iStar for learners? (MQ2) What are the challenges in applying iStar for learners, especially in iStar modeling? (MQ3) What are the challenges in analyzing iStar for learners? Do they know the strengths and limitations of iStar compared with other modeling frameworks? (MQ4) Are there any other challenges in learning iStar for learners?
(RQ2) How can the iStar modeling process be improved to address the challenges in learning and practicing?	(MQ5) What are the challenges and limitations of practicing iStar for learners? (MQ6) Can these challenges mitigated by a tool? (MQ7) What requirements do you have to mitigate these challenges?

During the process, each learner undertook the tutorial, modeling, and interview sessions individually as a single, continuous sequence.

Participants selection. We targeted undergraduate and master's students majoring in computer science and software as our beginner participants. All the selected learners meet the following requirements. Firstly, they possess a fundamental knowledge of requirements engineering and are familiar with at least one requirements modeling approach, such as use cases and data-flow diagrams. Secondly, they have developed software that has been running for over half a year with more than 100 users, and they understand the complexity and importance of requirements engineering. As most students lack real-world engineering experiences, we anticipate a low number of qualified participants. We verified their experience through detailed questionnaires and by reviewing their software projects. This rigorous selection process was necessary to ensure that participants had an adequate background to contribute meaningful insights into the challenges of learning iStar. As for the lecturer participants, we targeted those with more than ten years of experience in teaching or practicing iStar modeling.

Finally, our case study ultimately involves ten learners and three lecturers. To protect participant privacy, we have omitted the names of all participants. We use code names from *B1* to *B10* to represent learners and *L1* to *L3* to represent lecturers. Among the learners, *B1*, *B2*, and *B3* are undergraduate students in computer science. The other seven participants were master's students in computer science at the time of our empirical study. Their software development experience is depicted in detail in Table 4 As for iStar lecturers, all three have over ten years of experience with iStar modeling. They have utilized iStar for various research interests over the past decade. *L1*'s research interest is in general-purpose requirements engineering, and he is dedicated to making iStar widely used in software requirements engineering. *L2*'s research interest is domain-specific requirements engineering, and *L2* is devoted to making it easier to use iStar in the specialized area of research, facilitating the learning curve of iStar modeling. *L3* is interested in requirements engineering education and would like to introduce iStar more effectively to learners.

iStar tutorial. After an intense discussion with iStar lecturers, we chose a widely used iStar tutorial slide [29], which is comprehensive and easy to learn. The iStar tutorial was conducted one-on-one, with a similar session duration. Learners could ask lecturers questions anytime during the tutorial until they have no more questions about iStar.

iStar modeling. After understanding iStar, the learners participated in an iStar modeling session, in which we could observe their operations and analyze their potential requirements in iStar modeling. We are concerned that using unfamiliar requirements as the modeling scenario during the session may disturb participants' concentration. Thus, we asked them to use the most popular software they have developed as the application scenario. We adopted a whiteboard approach in the modeling session and provided only one black and one red pen, along with enough A4-size white paper for modeling, the two most commonly used pens in daily life. Such a pure modeling approach enables participants to focus on the conceptual model and the iStar modeling process, rather than the various usability issues associated with modeling tools. We asked iStar learners to design iStar models, with the goal of analyzing the requirements they have met and identifying the requirements for future updates to their software. During the modeling process, learners could refer to the tutorial slides and ask questions to the iStar lecturers anytime. Similar to the tutorial session, there is still no time limit. The modeling session ends once the learners believe they have completed the modeling. Additionally, we analyzed the models created by learners during the modeling session, focusing on identifying common modeling mistakes, misunderstandings of iStar notations, and other challenges. We asked lecturers to review their models, correct any syntax mistakes or inappropriate usage, and emphasize the concepts they found confusing to help students better understand iStar concepts. This ensures iStar learners have sufficient understanding for the following interview. In addition, as a final result of the modeling session, we require each learner to provide an iStar model that they are satisfied with.

Interview. The interview is the central part of the empirical study. Referring to Ournani et al. [30], we chose a semi-structured interview for our interview session. Semi-structured interviews require the main questions to be identified first, allowing participants to express their experiences freely while ensuring that key topics are covered. During the interview, we could ask follow-up questions at any time to obtain more detailed information based on the participant's responses to the main questions. We expect the views in interviews to be able to answer our research questions set in Section 4.1, so we designed our main questions (Table 2 for learners, Table 3 for lecturers) in the interview session according to our research questions. We interviewed learners and lecturers individually, based on these main questions, recorded the interviews, and took down their views. The interviews continued until we stopped, having obtained no new information or viewpoints from participants [31]. We use the denaturalism approach [32] for transcription, focusing on the content of the interviews while transcribing with less effort. Other studies, such as [30] and [33], also employed this method to transcribe their interview data. Meanwhile, it was as complete and credible as other methods, such as the verbatim method [32]. After we completed the transcription, we confirmed the records with them to ensure that what we had done was consistent with the views they wanted to express.

Result analysis. We analyzed the results gathered from the previous steps of our empirical study, including descriptive statistics and answers to research questions based on interview transcripts. Furthermore, to elicit the requirements and construct our modeling process, we applied additional techniques chosen from grounded theory [34], a theory developed by Strauss and Glaser, to obtain and form a new theory systematically rather than validate existing theories. Specifically, we analyzed these data based on the methodology proposed by Strauss and Corbin [31] for helping researchers based on the coding paradigm.

Due to practical constraints, specifically a limited number of participants and a fixed schedule, our data collection and analysis phases were conducted sequentially instead of iteratively. As a result, rather than fully utilizing the iterative procedures of constant comparison and theoretical sampling, we primarily focused on the analytical techniques of grounded theory, which include open, axial, and selective coding, according to the grounded theory guidelines proposed by Stol et al. [35].

Moreover, to address potential validity threats due to reliance on interviews alone, we enhanced the theoretical robustness of our study by performing data triangulation. Specifically, we cross-validated our findings through three distinct but complementary data sources: semi-structured interviews, analysis of participants' iStar modeling artifacts, and a comprehensive literature review focused on existing studies identifying difficulties and challenges in iStar modeling and existing iStar modeling tools. By comparing and synthesizing findings across these diverse data sources using a coding paradigm, we aimed to increase the depth, completeness, and reliability of our emergent theory.

5. Results and analysis

In this section, we present the analyzed results of the interviews with participants and answer the research questions formulated in Section 4.1 based on these results. We perform the iStar tutorial and modeling session for iStar beginners and record corresponding results based on protocol. Then, we use grounded theory to establish a systematic theory about difficulties in iStar learning and elicit corresponding requirements from our interview results. Finally, we analyze the threats to the validity of our empirical study.

Table 4
Software developed by iStar learners.

ID	Developed software	Developing time	Lines of code	Tech stack	Iterations	Running time	User scale	Requirements satisfaction
B1	An online modeling platform for teaching software engineering	1 yr.	Near 5000	ejs Express Node.js	Keep iterating	2 yrs.	Near 200	Completely satisfied
B2	An online judge system for teaching data structures	1 yr.	Over 10000	Echo Go	Major version with bug fixes	2 yrs.	Near 300	Completely satisfied
B3	Developed the online judge system with *B2*	1 yr.	Over 10000	Vue.js Echo Go	Keep iterating	2 yrs.	Near 300	Completely satisfied
B4	A serious game for teaching requirements analysis	3 mos.	Near 1000	Unity 3D C#	Keep iterating	1 yr.	Near 100	Moderately satisfied
B5	A web application providing information on campus life	1 yr.	Over 10000	Webview Python	Keep iterating	2 yrs.	Near 400	Completely satisfied
B6	A map application with Beijing's monuments and attractions	3 mos.	Near 5000	Vue.js Spring boot	Keep iterating	6 mos.	Near 100	Moderately satisfied
B7	A social engineering attacks detection system	6 mos.	Near 5000	React	Keep iterating	1 yr.	Near 100	Moderately satisfied
B8	An automatic generation system of social engineering attack defense strategy	6 mos.	Near 5000	Java Spring boot	Major version with bug fixes	1 yr.	Near 100	Moderately satisfied
B9	An ordering and delivery application for campus	1 mo.	Near 1000	Python Express Node.js	Only iterative bug fixes	6 mos.	Near 100	Moderately satisfied
B10	An anomaly modeling behavior detection platform	1 mo.	Near 1000	Qt Python	Major version with bug fixes	6 mos.	Near 100	Moderately satisfied

ID = Participant code, B = Learners

5.1. Modeling results

We perform the iStar tutorial and modeling session for iStar beginners and record corresponding results based on protocol. In this subsection, we present the basic picture of the modeling results and analyze the common types of modeling errors and issues.

5.1.1. Descriptive statistics of modeling results

Table 5 shows the tutorial and modeling results of the beginners. As planned, the requirements of the iStar models are consistent with the software they developed. The tutorial and modeling session can help beginners provide more credible and helpful information for the following interview. In the modeling session, beginners' modeling time ranges from 20 to 90 min, with an average of 46 min spent on modeling. The largest model contains 91 elements, the smallest contains 10, and the average is 47. After they finish modeling, we check and correct the syntax errors to make them learn from their mistakes. Since all the models reflect actual software requirements, they are not toy models, no matter how many elements they contain.

B5 built the most miniature models and took the shortest modeling time. Due to its single function, the size of the software itself is the smallest, so the final model is also the smallest in size. It is worth noting that the web application developed by *B5*, which serves campus life, has the most users.

The online judge platform developed by *B3* and *B2* was the most complex software system in this empirical study. Thus, the iStar model constructed by these two participants had the most model elements. Specifically, although *B3* and *B2* developed the software system together, they participated in our empirical study separately. *B2* took 30 min longer to model than *B3*, with a finer granularity of the actors' intentions in their scenario leading to more intentional elements and refinements. That is why the final model *B2* contains more elements than *B3*.

Refinements between intentional elements were most commonly used in the models finalized by beginners. The highest model elements in each beginner's model were the number of intentional elements reflecting the goals, tasks, and required resources of actors and the number of refinements were the highest model elements in each beginner's model. In the models from beginners, there were about three actors. The dependencies between actors ranged from two to twelve, with an average of seven dependencies. It is worth noting that all the beginners did not use is-a or participates-in relationships between actors during modeling. One possible reason is they can complete the requirements modeling around several specific roles.

5.1.2. Common types of modeling errors

There are several types of errors we found in the first version of the beginning modeling results: conceptual misinterpretations, dependency misuse, incomplete refinements, semantic ambiguity, and model scalability and complexity. This subsection will discuss these types of errors in detail.

Conceptual Misinterpretations. The analysis highlighted several frequent conceptual misinterpretations. For instance, learners exhibited considerable difficulty distinguishing between "goals" and "tasks", a key aspect of iStar's intentional element taxonomy. Among the ten learners, seven (B1, B2, B4, B6, B7, B9, and B10) incorrectly labeled tasks as goals or vice versa at least once. An illustrative example was seen in the model of participant B4, who mistakenly represented the action "Process Order" as a goal instead of a task.

Dependency Misuse. Dependency relationships posed another significant issue. Six participants (B2, B3, B5, B7, B9, and B10) inaccurately defined dependencies either by misrepresenting the direction or by misunderstanding the roles of depender and dependee. Participant B7, for example, incorrectly specified the actor "User" as dependee instead of depender in a "Feedback Submission" dependency, reversing the intended meaning of the relationship.

Incomplete Refinements. Incomplete refinement of intentional elements was observed in half of the models (B1, B3, B4, B6, and B8), reflecting difficulties in breaking down complex requirements into detailed subtasks. For instance, participant B3's model stopped refinement prematurely, leaving the goal "Improve User Satisfaction" inadequately detailed, thus reducing clarity and practical utility.

Semantic Ambiguity. Semantic ambiguity, arising from vague descriptions of intentional elements, was common in four learners' models (B2, B5, B9, and B10). This ambiguity often led to multiple interpretations, complicating model analysis and usage. For instance, participant B5 labeled a task simply as "Check Data", without specifying what data was being checked or why, causing confusion during model evaluation.

Model Scalability and Complexity. Participants also encountered issues related to model scalability and complexity management. Larger models, particularly from participants B2 and B9, exhibited layout difficulties, hindering readability and interpretation. The complexity of connecting numerous elements resulted in cluttered diagrams, as exemplified by B9's model, making it challenging to trace relationships and evaluate model coherence effectively.

5.2. Interview results

In this subsection, we answer research questions based on the views expressed by the ten learners and three lecturers.

5.2.1. Answering RQ1

RQ1: What specific challenges do beginners face when learning iStar at different cognitive levels, particularly understanding, applying, and analyzing?

According to the revised Bloom's taxonomy [14], we categorize the challenges that learners face when learning iStar into three cognitive levels: understand, apply, and analyze.

Understand. At the taxonomy's understanding level, we focus on comprehension of the modeling elements, semantics, and concepts of iStar modeling.

The main difficulties learners encounter are in mastering the conceptual model of iStar and expressing its semantics. Learning various concepts and distinguishing among them can be challenging for learners. As *B6* said, "I don't know which type of relationships should be used to connect elements. Dependencies are also confusing at first. These concepts are complicated to learn". *B2*, *B4*, and *B7* clarify the confusion regarding distinct types of element, such as softgoals versus goals, and the differentiation between agents and roles. *B9* and *B10* point out that iStar has various element types to be learned, making distinguishing and applying these concepts difficult. *B7* supports this view, thinking that more element types lead to a higher learning curve. Learners need to master many element types before conducting iStar modeling fluently. However, *B8* argues that there are not many element types in iStar, and thus learning iStar is not hard. Nevertheless, more learners still think that the large number of iStar element types and the complexity of conceptual analysis confuse them and make their modeling results error-prone.

Apply. At the Applying level, we concentrate on using iStar to model requirements effectively and mastering the associated methodologies and best practices.

From the perspective of lecturers, an effective way to introduce iStar to learners could be to encourage them to engage in trial and error. However, *L3* points out that many learners lack an effective modeling process due to unclear direction or overemphasis on unnecessary details, making iStar modeling hard to apply. Most learners naturally move from abstract to concrete during iStar modeling, including *B2*, *B3*, *B4*, *B6*, *B7*, *B9*, and *B10*. They use a progressive process, first listing actors and then analyzing each one's intentions. Moreover, *B1* and *B2* propose an abstract-to-concrete approach, first analyzing the user requirements, i.e., actors who raise more requirements, and then eliciting the system requirements, i.e., actors who implement the requirements. In addition, *B5* mentions that the modeler's knowledge of the requirements influences the choice of the iStar modeling process. Only if a modeler comprehensively understands the requirements, top-down modeling can be used. Otherwise, top-down modeling can be very challenging for iStar learners. For example, *B8* supports a top-down modeling scheme for refining intentions. Users could define actors' goals first and then refine these goals into multiple tasks. However, *B8* admits that this approach requires a complete and clear understanding of the modeling objectives. Otherwise, a bottom-up modeling process from partial to whole is more appropriate.

Another difficulty in applying iStar is expressing complex natural language requirements using iStar notations. As *L1* points out, many practical scenarios are pervasive in software projects but difficult for learners to model. *B1* tells that there are often multiple ways to refine requirements into intentional elements, and the abstraction choice affects model relationships, which makes it hard to find the proper abstraction pattern when performing iStar modeling.

Analyze. At the Analyzing level, we emphasize understanding iStar's strengths and limitations and how to leverage it for model analysis. All ten learners agreed that iStar is more complex than other models like Flowcharts and UML diagrams. Models that are easy to use have a lower level of abstraction. For instance, *B5* mentioned that UML class diagrams and flowcharts are more intuitive as they are abstractions of natural objects. Thus, iStar's higher level of abstraction makes it more

Table 5
Modeling results from beginners.

	Time spent (in minutes)		Number of model elements				
ID	Tutorial	Modeling	Actors	Intentions	R.	D.	SUM
B1	20	50	4	20	21	8	53
B2	26	90	4	42	33	12	91
B3	26	65	4	30	20	10	64
B4	25	90	3	16	11	3	33
B5	23	20	2	11	9	2	24
B6	22	45	2	15	13	1	31
B7	25	34	2	33	28	2	65
B8	23	25	1	5	4	0	10
B9	24	30	2	37	31	0	70
B10	26	14	1	13	10	0	24
Mean	24	46.3	2.5	22.2	18	3.8	46.5

ID = Participant code, B = Beginner, R. = Refinements, D. = Dependencies

challenging to learn. *B4* encountered difficulties distinguishing the hierarchy of intentional elements, stating "find it difficult to differentiate between the hierarchy of intentional elements" Similarly, *B2* agreed that abstract thinking presents the greatest challenge, noting "Thinking abstractly about the modeling elements is the hardest part, especially when compared to more concrete diagrams like flowcharts".

Learning iStar can be challenging due to differences from other frameworks, resulting in a steep learning curve. *B2* and *B9* noted significant variations in approaches between iStar and other modeling frameworks. As a result, many issues need to be considered, which can be discouraging and lead to a perception that iStar is complicated. *L3* suggests that learners struggle to use iStar effectively because they fail to recognize its unique strengths and limitations, possibly due to their familiarity with other models. "Learners often overlook the unique strengths and limitations of iStar simply because they're accustomed to thinking in terms of more traditional models".

5.2.2. Answering RQ2

RQ2: How can the iStar modeling process be improved to address the challenges in learning and practicing?

In addition to the challenges of understanding, applying, and analyzing cognitive levels in iStar learning, several practical limitations can make iStar practice difficult.

First, the quality of the artifact being analyzed is important. *B2*, *B3*, *B4*, *B5*, and *B10* mention the impact of requirements quality on modeling difficulty. *B4* states that the quality of the requirements is as important as the quality of the requirements analysis itself.

Moreover, preferences for laying out elements and modeling views, as well as the time and effort required to modify model elements and relations, can pose challenges. *L1* points out that a complex model layout can hinder subsequent modeling as the number of elements increases. *L3* observes that incorrect element placement forces learners to adjust positions, disrupting their modeling process frequently. *L1* and *L2* suggest that the modeling process often involves many additions of similar elements and relationships, making manual addition inefficient and prone to omissions.

To mitigate all these challenges, most of our participants agree that these difficulties can be solved by a tool. The lecturers generally believe that iStar modeling tools can effectively alleviate some of the challenges mentioned earlier. However, there is some disagreement among learners. *B1*, *B4*, *B6*, *B8*, *B9*, and *B10* agree that modeling tools would alleviate some difficulties they encountered. Several participants highlighted the benefits of modeling tools for improving their learning experience. B1 emphasized the need for automated syntax checking to help beginners identify and correct errors quickly, reducing cognitive load and allowing focus on conceptual modeling. B4 noted the inefficiency of adding model elements manually and suggested batch addition features to streamline the process and minimize errors. B6 stressed the importance of auto-layout functionality for organizing elements and improving model readability, especially in complex projects. B8 advocated for tools with predefined templates and example models to simplify understanding of iStar's element types and provide structured guidance. B9 highlighted the value of feedback and hints for model refinement, particularly to address incomplete refinements or ambiguous dependencies. Lastly, B10 suggested automatic dependency analysis to visualize relationships, identify errors, and enhance modeling accuracy. These features collectively support learners by reducing manual effort, preventing errors, and improving their understanding of modeling concepts. More radically, *B2* and *B7* believe modeling tools can greatly alleviate various difficulties. *B5* also agrees on the positive side of modeling tools to alleviate the difficulties. However, *B5* also mentions that modeling tools might introduce more difficulties when the tools are not designed well. We believe this point is critical and deserves the attention of developers. In contrast, *B3* does not think tools can help alleviate iStar modeling. *B3* takes a more narrow view of the application of the tool; he states that the difficulties of iStar modeling mostly come from the need for more understanding of the concepts of iStar. It can be alleviated by more practice and comprehension of iStar modeling rather than modeling tools. Although *B3* disagrees with the direct role of tools in alleviating the modeling difficulties, *B3* agrees that tools have benefits to the practice of modeling, which may indirectly help the difficulties. For example, *B3* thinks a modeling tool that supports automatic layout could provide a clearer view, allowing modelers to focus more on modeling ideas without being bothered by model layouts.

To address these challenges and improve iStar's learning and practice, we propose strategies organized into four key areas: understanding, applying, analyzing, and mitigating practical limitations. In terms of understanding iStar modeling, providing technical help such as iStar concepts and element types is necessary. Besides, automated syntax checks and corrections can help learners better understand iStar element types. *B10* suggests the system should provide automated model correction suggestions, including incorrect element and relationship types. In terms of applying iStar modeling, technical help and guidance for modeling are essential. For instance, *L3* suggests providing detailed guidance on the modeling process. For example, heuristic questions to facilitate the modelers' thinking and help them decide the starting point of the modeling analysis. *L1* and *L2* specifically propose the need for a standard template library (S3) in iStar modeling. To enhance the analyzing aspect of iStar models, advanced visualization and interaction features are necessary, including automatic layout for models, organizing lines between elements, allowing free adjustment of element positions, and providing different views for the same model. *B2*, *B3*, *B4*, *B7*, *B8*, and *B10* mention that auto layout is an essential feature for modeling tools. All lecturers have also required this requirement. According to them, this feature can provide a better visual experience for iStar modelers. Moreover, *B5* suggests that this tool can also provide different views for different roles in enterprise software application development teams. Addressing practical limitations is crucial for improving iStar modeling. Streamlining the modeling process by batch-adding model elements and relations can significantly reduce the cost of editing models. All lecturers have approved this requirement. Furthermore, *L1* and *L2* believe a modeling process that supports automated modeling from requirements documents to iStar models could be beneficial. *B5* also proposes the requirement for automated modeling, desiring a modeling tool that could support modeling from requirements documents to iStar models. To enhance the modeling process, *B6* proposes that specific prompts can be shown in the modeling process, such as the types of elements suitable for relationships and the placement locations for these elements. *B6* also believes that subsequent modeling operations can be predicted and recommended by the modeling tool.

While modeling tools can streamline certain aspects of iStar practice, it is crucial to recognize their limitations. Tools should be designed to complement educational efforts, providing support without substituting the fundamental learning of concepts. As B5 noted, the effectiveness of tools is contingent on thoughtful design that aligns with learners' needs.

After analyzing the corresponding interview results, a literature review was conducted to analyze existing studies on these modeling difficulties in the following subsection.

5.3. Literature review

This literature review examines the current state of research related to iStar modeling, focusing on two key aspects: the challenges encountered by modelers when using the iStar framework and the tools developed to support iStar modeling activities. By analyzing existing studies on modeling difficulties faced by novices and non-technical stakeholders, along with evaluating the capabilities and limitations of available modeling tools, we aim to identify gaps in current approaches and establish a foundation for our proposed enhancements.

5.3.1. Challenges of iStar modeling

Several studies have addressed challenges related to scalability, complexity, and conceptual difficulties encountered by novices when modeling using the iStar framework. Horkoff [36] conducted an observational study highlighting significant conceptual ambiguities faced by novice modelers, notably graduate and undergraduate students. Specific issues included difficulties in distinguishing between dependencies and clearly identifying abstract actors. Beginners frequently struggled to discern whether an element should be modeled as an external dependency or an internal intentional component, leading to ambiguity in modeling internal structural relationships. Additionally, novices expressed uncertainty regarding the effectiveness of modeling tools in mitigating these challenges, primarily due to their limited intuitive grasp of the underlying concepts. To address these difficulties, this study recommended initiating modeling activities with hybrid SD/SR models, emphasizing initial understanding of participants, system boundaries, and dependency relationships prior to tackling internal structural complexities.

Further addressing conceptual complexity, Estrada et al. [37] conducted empirical evaluations of iStar within industrial contexts, highlighting its capability to capture organizational and social intricacies alongside notable limitations. A significant challenge identified was the framework's inadequate support for detail refinement and modularization, particularly as models increased in scale and complexity. The absence of clear modular mechanisms significantly constrained scalability, complicating model management and increasing maintenance overhead. These findings underscore the importance of augmenting iStar with advanced modularization and detail refinement techniques to better support extensive, real-world system scenarios.

Carvallo and Franch [38] explored iStar usability issues specifically from the perspective of non-technical stakeholders. Their research indicated that stakeholders commonly faced difficulties accurately specifying dependency types and directions, resulting in ambiguities during dependency refinement. Additionally, they observed frequent neglect in properly expressing non-functional requirements within models, thus leaving gaps in comprehensive system perspectives. This study advocated against exclusive reliance on specialized tools, recommending instead the provision of context-driven examples and structured guidelines, combined with comprehensive training tailored to stakeholders' intuitive understanding and practical needs.

These challenges highlight several key areas for improvement in iStar modeling. In terms of framework capabilities, there is a need for enhancements that allow for a comprehensive expression of non-functional requirements and support effective modularization for complex systems. From a strategic perspective, modelers would benefit from techniques that help to: differentiate between dependencies and internal intentional elements; clarify the types and directions of dependency relationships; distinguish between different actor abstractions such as agents and roles; incorporate advanced modularization approaches for improved scalability; identify and define abstract actors within system descriptions; and systematically capture and represent non-functional requirements in the models.

5.3.2. iStar tools

Several tools have been developed to support iStar modeling, each with its unique features. The iStar wiki has already enumerated current mainstream tools [39], including detailed explanations of their various features.

Additionally, Li et al. conducted research on the challenges and tradeoffs of iStar tools, performing case studies on three tools, including OpenOME [40], MUSER, and Leaf [41], to establish goal models to capture the collected knowledge.

In the following, we introduce several classic iStar modeling tools. REDEPEND [42] leverages the graphical modeling capabilities of MS Visio to provide an intuitive interface for creating Strategic Dependency and Strategic Rationale models. It incorporates usability enhancements such as easy navigation between related model elements, synchronization across multiple views, and model-checking features to enforce modeling rules. Notably, REDEPEND automates the generation of textual requirements from iStar models using predefined patterns, significantly aiding requirements documentation and validation processes in real-world industrial applications.

J-PRiM [43] supports the PRiM methodology, offering a structured approach to process reengineering through iStar. Built using Java and Eclipse frameworks, J-PRiM features a database-backed approach to managing model elements. The tool guides users through phases such as actor identification, model generation, evaluation of process alternatives, and trade-off analysis. Uniquely, J-PRiM represents iStar models in a hierarchical, tree-based format rather than traditional graphical displays, aiming to improve the scalability and manageability of large-scale models.

In recent years, more modern iStar modeling tools have emerged. Creative Leaf [41] is a web-based tool that integrates creativity techniques with iStar goal modeling for requirements engineering. The tool addresses the challenge that stakeholders often struggle to articulate goals or envision innovative solutions beyond their existing experiences. This tool incorporates a creativity panel enabling structured techniques that support both divergent creativity that generates ideas and convergent creativity that selects and develops ideas. Unlike traditional creativity workshops requiring skilled facilitators, Creative Leaf provides a systematic framework that makes creativity techniques accessible while directly connecting creative ideas to goal models, helping capture non-obvious requirements that could lead to innovative systems with competitive advantages.

piStar [44] is an online modeling tool compliant with the iStar 2.0 standard, designed with simplicity and extensibility in mind. It prevents common modeling mistakes by disallowing invalid links such as dependency links from an actor to itself or incorrect contribution links. It supports extensibility through JavaScript and offers REST-based web

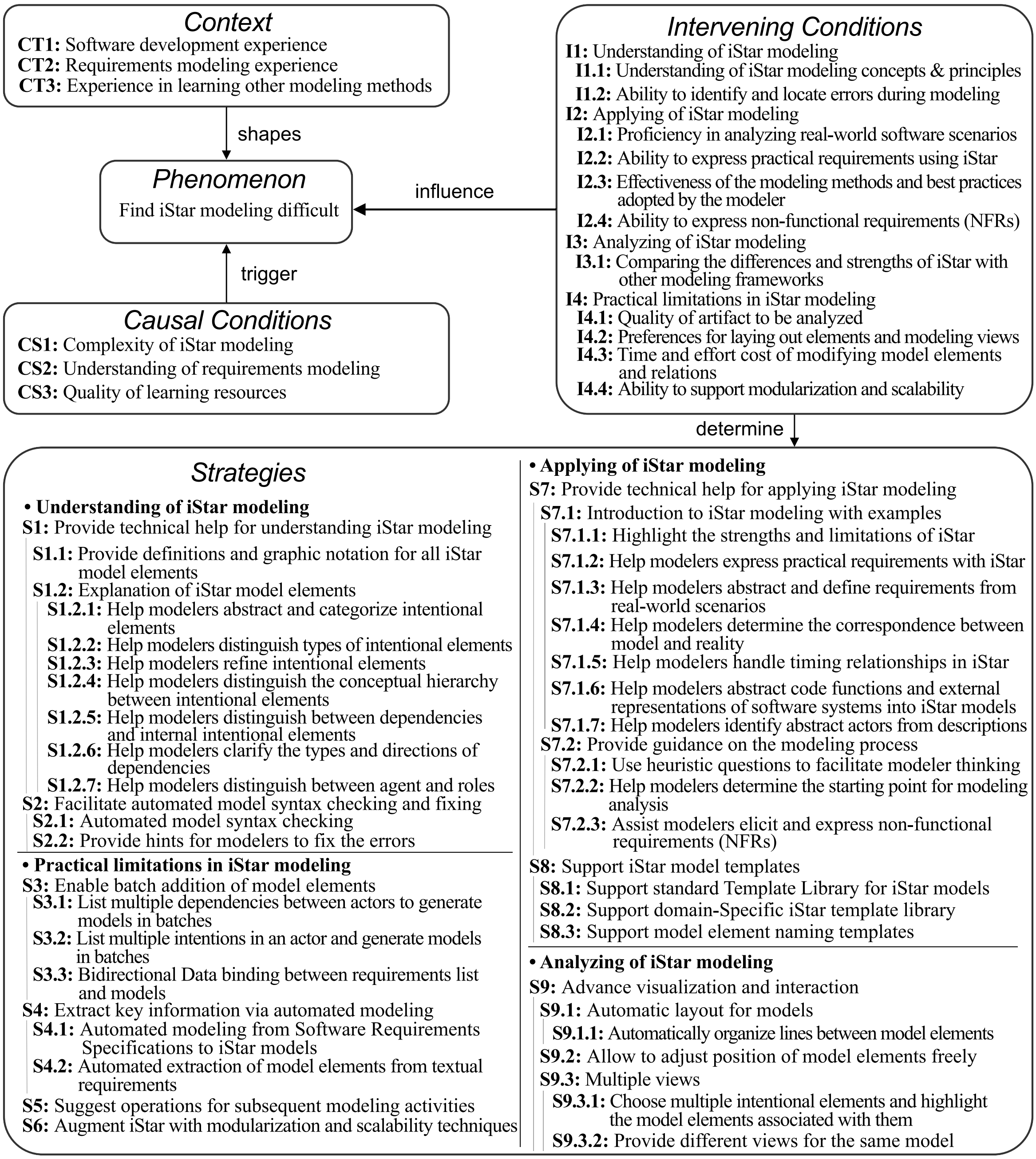

Fig. 3. Using Coding Paradigm to structure our theory.

service integration for server-side processing, making it accessible for developers to extend functionality in various programming languages like Java and Python.

Overall, these tools demonstrate varying levels of support for automatic error prevention, extensibility, usability enhancements, and integration with requirements elicitation and documentation tasks. A comprehensive overview of these and other iStar tools can be found on the iStar wiki, which offers detailed comparisons of available tools based on numerous functional criteria.

The next section details our implementation of these theoretical enhancements into a structured coding schema, explicitly designed to operationalize these insights for practical use.

5.4. Coding and theory structuring

After interviewing, we transcribe the records according to our research protocol. Based on these transcripts, we perform grounded theory techniques to establish a theory of the difficulties and requirements for learners in iStar modeling. There are different procedures in the codification process of grounded theory: open coding, axial coding, and selective coding [31]. We conduct these procedures in an interweaving way. In this study, we utilize NVivo to perform our grounded theory. Nvivo is a widely utilized software package for qualitative research, which supports systematic coding, organization, and analysis of textual and multimedia data, enhancing the rigor and transparency in qualitative methodologies, including grounded theory approaches [45]. As for our empirical study, there are two different interviews for iStar learners and lecturers, each including multiple data collection and analysis iterations. In each iteration, we add the transcribed data to Nvivo data analysis software and apply different coding procedures to analyze these materials.

In terms of the coding process, all researchers collaboratively participated in the initial open coding stage to ensure comprehensive familiarity with the data and consistent generation of preliminary concepts. During the axial coding phase, the coding tasks were distributed among the researchers, who independently coded specific subsets of the data. Regular meetings were scheduled for researchers to review, discuss, and consolidate their coding results. When disagreements occurred, they were resolved through detailed group discussions until a consensus was reached. The final selective coding phase was conducted jointly by the entire research team, thereby ensuring the consistency and coherence of the resulting theoretical structure.

5.4.1. Open coding

We conduct open coding to conceptualize these materials. First, we review transcripts line by line and tagged phrases with a set of codes consisting of a few words to represent them. Subsequently, similar codes are grouped into concepts, and related concepts are integrated into categories. During the entire coding procedure, we keep writing memos to record the emerging concepts and categories for their relationships and constantly compare these memos and codes to generate new concepts and categories.

For example, from the transcript, "Learning other modeling does not help learn iStar, and may even have a negative impact because it is easy to confuse the concepts of the relationship between the two at the beginning", it can be inferred that the discussion is about the relationship between learning iStar modeling and other modeling methods. Therefore, we annotate this content as the code "Experience in learning other modeling methods". In addition, this passage also discusses a specific difficulty in learning iStar, which is the confusion of relation concepts in iStar. Thus, we annotate this as the code "Learning relation concepts". In subsequent analysis, this code is grouped into the concept named "Learning iStar syntax" and finally "Understanding of iStar modeling concepts & principles". This way, we apply the open coding procedure on the transcripts and yield corresponding codes, concepts, and categories.

5.4.2. Axial coding

After we annotate codes and concepts, we interweave them in axial coding to continuously identify relationships between codes and categories. Specifically, we use the coding paradigm in the axial coding procedure to facilitate the analysis. The coding paradigm contains characteristics such as phenomenon, context, causal conditions, intervening conditions, strategies, and consequences to map our categories into a systematic structure. Besides, these characteristics can generate the core category in the subsequent selective coding procedure [31]. This procedure involves mapping our concepts and categories from open coding to these coding paradigm terminologies. It is noted that consequences in the coding paradigm are omitted, as it is the purpose of our study. We gather the essential requirements and other ideas for overcoming learners' and lecturers' difficulties and categorize them as strategies. We divide these strategies into four sub-categories, with multiple concepts included. Some requirements focus on providing help for iStar modeling, while others focus on automated modeling and other methods to support iStar modeling.

5.4.3. Selective coding

In selective coding, we analyze the dynamics from the coding paradigm and build a storyline to present the structure of the difficulties and requirements for learners to learn and practice iStar modeling. During the selective coding procedure, the theory reaches theoretical saturation, which means that these results fit these materials and no new concepts, categories, or insights emerge. We treat strategies as the main category of our theory. By expressing and resolving them, strategies can relate to every other concept and category.

5.4.4. Theory details

Fig. 3 presents our theory using Strauss and Corbin's coding paradigm. We can see that the phenomenon of finding iStar modeling difficult is influenced by related contexts, causal conditions, and intervening conditions.

Moreover, Table 6 shows how these intervening conditions and strategies align with insights gathered from participants and literature reviews. The columns represent the perspectives of individual iStar learners and lecturers, as well as findings from the literature review. A checkmark indicates when a participant's viewpoint from the interview or a point from the literature review supports or aligns with the respective condition or strategy. This visualization offers a clear overview of how different challenges and proposed solutions resonate across participants and existing research, thereby reinforcing the validity and applicability of our findings.

To investigate the phenomenon of finding iStar modeling difficult, we employ a grounded theory-based coding paradigm to analyze various influencing factors and their interrelationships.

Context. We considered contexts such as software development experience (CT1), requirements modeling experience (CT2), and experience learning other modeling methods (CT3). These contextual factors shape learners' perceptions of iStar's difficulty, thereby influencing the phenomenon.

Causal conditions. The causal conditions section encompasses the complexity of iStar modeling (CS1), the understanding of requirements modeling (CS2), and the quality of learning resources (CS3). These causal conditions directly contribute to the challenges experienced by modelers when learning and applying iStar. The complexity of iStar modeling and the lack of understanding of requirements modeling can exacerbate the difficulties faced by modelers. Additionally, the quality of learning resources available to modelers can significantly impact their ability to grasp iStar concepts and effectively apply them in practice.

Intervening conditions. The intervening conditions section plays a crucial role in explaining why iStar modeling is difficult. These conditions are categorized into four main aspects: understanding iStar modeling (I1), applying iStar modeling (I2), analyzing iStar modeling (I3), and practical limitations in iStar modeling (I4).

Under the understanding of the iStar modeling aspect (I1), conditions include the understanding of iStar modeling concepts and principles (I1.1) and the ability to identify and locate errors during modeling (I1.2). These factors influence the modelers' comprehension of iStar and their ability to utilize it effectively.

Besides, the application of the iStar modeling aspect (I2) encompasses several conditions, such as proficiency in analyzing real-world software scenarios (I2.1), the ability to express practical requirements using iStar (I2.2), the effectiveness of the modeling methods and best practices adopted by the modeler (I2.3), and the ability to express non-functional requirements (I2.4). These conditions determine how well modelers can translate real-world requirements into iStar models and apply modeling techniques.

Moreover, the analysis of the iStar modeling aspect (I3) includes conditions like comparing iStar's differences and strengths with other modeling frameworks (I3.1). This condition influences the modelers' understanding of iStar's characteristics and how it differs from other modeling approaches with which they may be familiar.

Lastly, the practical limitations in the iStar modeling aspect (I4) cover conditions such as the quality of the artifact to be analyzed (I4.1), the preferences for laying out elements and modeling views (I4.2), the time and effort cost of modifying model elements and relations (I4.3), and the ability to support modularization and scalability (I4.4). These practical constraints can significantly impact the ease and efficiency of the iStar modeling process.

Strategies. To address these challenges and improve iStar learning and practice, we propose strategies based on results.

Table 6
Mapping of intervening conditions and strategies to data sources.

Concept	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	L1	L2	L3	LR
Intervening Conditions (I)														
I1.1		✓		✓		✓	✓		✓	✓		✓		✓
I1.2	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓		✓
I2.1		✓	✓	✓	✓	✓	✓	✓	✓	✓			✓	
I2.2	✓				✓			✓			✓			
I2.3		✓	✓	✓	✓	✓	✓	✓	✓	✓			✓	
I2.4		✓		✓							✓		✓	✓
I3.1		✓			✓	✓				✓			✓	
I4.1		✓	✓	✓	✓					✓		✓		
I4.2		✓	✓	✓			✓	✓		✓	✓		✓	
I4.3				✓							✓	✓		
I4.4												✓	✓	✓
Strategies (S)														
S1		✓		✓		✓	✓		✓	✓				✓
S2	✓									✓	✓	✓	✓	✓
S3				✓							✓	✓		
S4					✓						✓	✓		✓
S5						✓					✓	✓		
S6		✓							✓			✓	✓	✓
S7						✓					✓	✓	✓	✓
S8								✓			✓	✓		
S9		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

*LR = Literature Review

We acknowledge that during interviews, participants suggested numerous strategies and requirements, some of which raised concerns regarding their practicality or clarity. To ensure the strategies included in our final theory were valid and useful, we engaged the three lecturers in a dedicated post-interview discussion. In this session, we presented all initially identified strategies and requirements to the lecturers, who evaluated each according to its relevance, feasibility, and clarity. Requirements or strategies that were mentioned by fewer than two participants or deemed impractical by lecturers due to ambiguity, excessive complexity, or limited applicability were excluded. For instance, a strategy involving advanced machine-learning-driven automation of entire modeling processes was suggested by only *B5*, and lecturers considered it excessively complex for beginners, thus resulting in its exclusion. Another discarded strategy was the real-time collaborative modeling feature, which, despite being interesting, was considered beyond the current study's scope and impractical for initial iStar learning contexts. By employing this structured filtering approach, we ensured that the finalized set of strategies effectively addresses genuine learning challenges while maintaining clarity and practicality for educational use.

These strategies are organized into four key areas: understanding, applying, analyzing, and mitigating practical limitations.

For understanding iStar concepts and element types, technical help is crucial (S1). This includes providing clear definitions and graphic notation for all iStar model elements (S1.1) and explanations of intentional elements (S1.2). Modelers should be assisted in abstracting and categorizing intentional elements (S1.2.1), distinguishing between goals and tasks (S1.2.2), refining intentional elements (S1.2.3), understanding the conceptual hierarchy between them (S1.2.4), distinguishing between dependencies and internal intention elements (S1.2.5), clarifying the types and directions of dependencies (S1.2.6), and distinguishing between agents and roles (S1.2.7). Automated model syntax checking (S2.1) and hints for fixing errors (S2.2) can further facilitate the understanding process.

In terms of applying iStar in modeling practice, technical help and guidance are essential (S7). This includes introducing iStar modeling with examples (S7.1), highlighting its strengths and limitations (S7.1.1), and helping modelers express practical requirements (S7.1.2), abstract and define requirements from real-world scenarios (S7.1.3), determine the correspondence between model and reality (S7.1.4), handle timing relationships (S7.1.5), abstract code functions and external representations into iStar models (S7.1.6), and identify abstract actors from descriptions (S7.1.7). Providing guidance on the modeling process (S7.2), such as using heuristic questions to facilitate thinking (S7.2.1), determining the starting point for analysis (S7.2.2), and assisting modelers in eliciting and expressing non-functional requirements (S7.2.3), is also beneficial. Supporting iStar model templates (S8), including a standard library (S8.1), domain-specific templates (S8.2), and naming templates (S8.3), can further assist in applying iStar effectively.

To enhance the analyzing aspect of iStar models, advanced visualization and interaction features are necessary (S9). These include automatic layout for models (S9.1), allowing free adjustment of element positions (S9.2), highlighting model elements associated with selected intentional elements (S9.3.1), and providing different views for the same model (S9.3.2).

Addressing practical limitations is crucial for improving iStar modeling. Enabling batch addition of model elements (S3), such as listing dependencies between actors (S3.1) or intentions in an actor (S3.2) to generate models in batches and provide bidirectional data binding between requirements lists and models (S3.3), can streamline the modeling process. Extracting key information via automated modeling (S4), including generating iStar models from software requirements specifications (S4.1) and extracting model elements from textual requirements (S4.2), can also alleviate practical challenges. Additionally, suggesting operations for subsequent modeling activities (S5) can guide modelers through the process. Moreover, iStar modeling lacks modularization and scalability, making it inconvenient and challenging to build large and complex models. Augmenting iStar with modularization and scalability techniques (S6) can help enhance its modeling performance.

5.5. Threats to validity

In this subsection, we analyze threats that might affect the validity of our study. While we understand the emphasis on qualitative validity criteria such as credibility, resonance, usefulness, and the extension of cumulative knowledge in the context of Grounded Theory, we have chosen to continue discussing the more classical validity threats as outlined by Wohlin et al. [28]. The reason for this is that we believe that classical validity threats, such as construct validity, internal validity, and conclusion validity, are still highly relevant to our case study.

Additionally, since our study only partially applied Grounded Theory techniques rather than fully adopting the entire methodological approach, we opted not to rely solely on the validity criteria typically associated with Grounded Theory studies. As a compensatory measure, we explicitly discussed the validity threats related to our applied Grounded Theory methods within the construct validity subsection.

Construct validity. Construct validity reflects the extent to which the research protocol meets the research objectives and research questions. In our study, the threats to construct validity can emerge in the protocol design, especially when measuring and collecting results. To mitigate this, we designed our empirical study protocol aligned with our research objectives to mitigate threats. Besides, we carefully designed the interview questions to directly address our research questions, guided by Bloom's taxonomy. We pilot-tested the interview protocol to ensure clarity and relevance. Additionally, we employed member checking by sharing the transcribed interviews with participants for validation, thus enhancing the credibility of the data collected. To combat communication vagueness, we encourage participants to think aloud during interviews and use main interview questions to keep our questions topic-focused. Moreover, we ensure data accuracy through transcription confirmation with participants and grounded theory-based coding and analysis.

One potential threat to construct validity in our study arises from the modeling method we adopted. Specifically, participants used a manual, hand-drawn approach on A4 paper with only black and red pens provided during the modeling sessions. This choice aimed to minimize external distractions and help participants concentrate solely on the conceptual aspects of iStar modeling. However, this method might influence participants' perceptions and requirements for tool support, as their experiences were not shaped by any existing modeling tools. Consequently, their identified needs for tool functionality and usability may differ compared to scenarios where modeling tools are involved. To mitigate this threat, we explicitly discussed and clarified the possible impact of our manual modeling approach on participants' perceived requirements and integrated insights from experienced lecturers and existing literature on existing iStar tools.

To further strengthen construct validity, it is important to emphasize that the coding paradigm employed in our study was systematically constructed according to the established guidelines by Strauss and Corbin [46]. By strictly following the recommended phases of open coding, axial coding, and selective coding, we ensured clarity and consistency in identifying, categorizing, and relating concepts. This rigorous and structured approach to data analysis helps to mitigate threats arising from subjective interpretation or coding inconsistencies, thus enhancing the reliability and robustness of the constructs derived from our qualitative data.

Internal validity. Internal validity requires researchers to consider the factors influencing research as comprehensively as possible. While our participant selection based on strict conditions rather than random sampling could potentially limit perspectives and threaten internal validity, it is a necessary constraint given our research objectives, focusing on selecting iStar learners with enough software engineering and development background to bridge the gap between student experimentation and industrial practice. Although these participants are beginners in iStar, their current level may not be enough for them to fully master the iStar framework. As a result, they may make errors and misunderstandings in both modeling and interviews. However, these mistakes have been valuable in highlighting the specific challenges learners face when trying to learn iStar. During the interviews, we paid close attention to the varied perspectives among the learners and engaged in detailed discussions to explore these differences. Moreover, the inclusion of iStar lecturers from various research fields in our participant pool broadens the range of experiences and perspectives, thus mitigating any potential bias from participants and enhancing the internal validity of our empirical study.

Additionally, a potential threat to our study's internal validity arises from using different modeling cases for each participant. We asked participants to use iStar modeling on their own software projects for authenticity and deeper engagement. While this enhances validity, it may lead to variability in how they perceive and represent iStar concepts, affecting consistency. We recognize that differing problems could influence participants' understanding and application of iStar. However, we prioritize capturing genuine experiences over uniformity. Our goal is to reveal the diverse challenges learners face when adopting iStar in real-world contexts. Future research may consider standardized cases to strengthen internal validity.

One threat to internal validity is the potential bias arising from all learners receiving training from the same lecturers, which might lead to similar viewpoints among participants. Although identical training may indeed influence learners to some extent, individual differences among learners can reduce this effect. While common training sessions might partially influence learner responses, they represent only one dimension affecting theoretical saturation in qualitative research. Future studies could mitigate this threat by involving learners from diverse cohorts who have experienced different training styles or instructors.

External validity. External validity reflects the ability to generalize the results of the study. Our study's main problem is whether our study can be generalized to fit real-world software practice. To promote external validity, we include learners with experience in actual software development and iStar lecturers experienced in teaching and promoting iStar in various fields. During the modeling session for iStar learners, these models established are relatively small. Even though learners may encounter more complexities with larger iStar models as the model size increases, their real-world software experience and requirements engineering knowledge can help ease these difficulties. Not to mention that iStar lecturers have plenty of iStar teaching or promoting experience, which can share their more profound understanding of learning difficulties to compensate for the omissions of learners. However, promoting learning and practicing iStar still needs more empirical experience to improve the usability and ease of use of iStar continuously.

Besides, a primary threat to the external validity of our study is the relatively small sample size, involving ten learners and three instructors. This limited number of participants may constrain the generalizability of our findings to the wider population of iStar learners and practitioners. Our research focuses on obtaining in-depth insights into the difficulties and challenges learners face when adopting iStar, drawing from their unique experiences and perspectives. While this approach provides valuable qualitative data, we acknowledge that it may not capture the full spectrum of challenges encountered by a larger, more diverse population. Enhancing the generalizability of our findings is a priority for future research, which will involve expanding the participant pool and including a broader range of backgrounds and experiences. Such efforts will help to validate our results and ensure that the strategies we propose are applicable to a wider audience.

Moreover, due to the limited number of participants, the proposed theory may reflect a state of local saturation. This possibility arises because the current saturation is based solely on the perspectives of the present group of learners, and it does not account for the emergence of additional or differing viewpoints from other learners or in different contexts. To address this limitation, we conducted a literature review to complement and enhance our theory by incorporating existing insights on the challenges encountered in iStar modeling.

Conclusion validity. Conclusion validity focuses on how sure we can be that our treatment in an experiment relates to our observed outcome. In this empirical study, we have involved ten learners and three lecturers, which may threaten the conclusion validity of the lack of participants. However, we systematically perform our proposed protocol and collect detailed results, which can mitigate the threats. Specifically, we employ triangulation by corroborating interview findings with model analysis and previous literature. Furthermore, we emphasized the importance of reaching consensus within our research team on the interpretation of data, particularly when discrepancies arose. Additionally, by comparing our results with findings from other

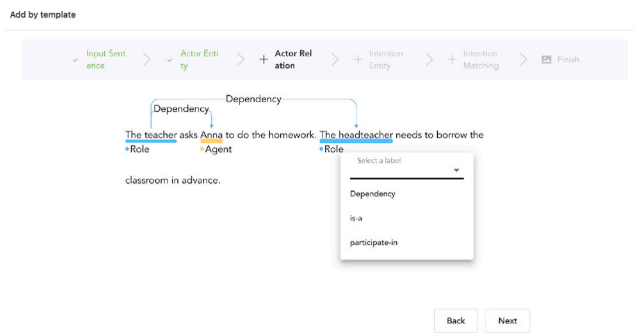

Fig. 4. The functionality of adding elements from SRSs.

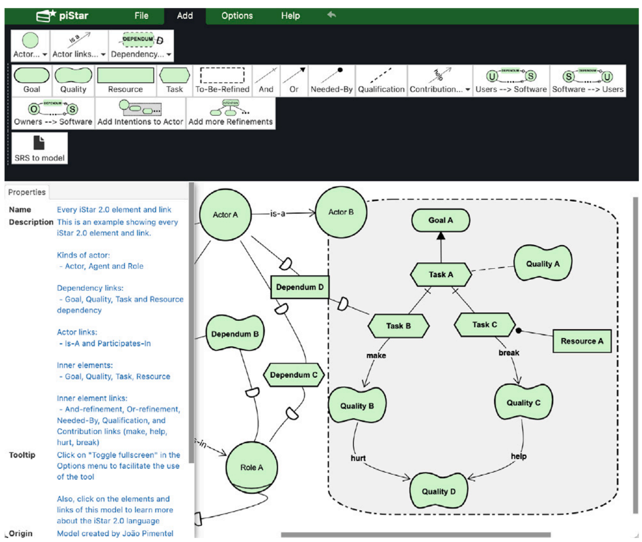

Fig. 5. The functionality of adding elements via templates.

studies in the field, we strengthened the depth and rigor of our generated theory, ensuring it is both reliable and comprehensive. We welcome future research efforts that can replicate our empirical study, expand the requirements we elicit, and thus strengthen our research.

6. A prototype tool

After constructing the theory of difficulties learners encounter in iStar modeling and gathering strategies as modeling requirements, we further implement a prototype tool based on the inspiration of our studies, aiming to promote iStar learning and practicing. We chose to develop our modeling tool based on piStar [44], one of the most popular open-source modeling tools in the iStar community. We develop a prototype tool,[1] which fulfills most of the requirements. We have also recorded a video demonstrating our prototype tool,[2] showing the features of the prototype tool.

To promote iStar learning and practicing, we implement some functionalities mentioned in the interview, such as multiple ways of adding model elements and auto-layout. The prototype tool also supports adding elements in batches, using templates, and adding elements from software requirements specifications (SRSs). Fig. 4 presents the functionality that extracts model elements from SRSs through a multistep process. It begins by inputting each sentence and then performs actor entity Extraction to identify roles and agents. Next, actor relation extraction determines dependencies between the actors. Finally, intention entity extraction matches the text to predefined intention labels. The extracted elements can then be added to the iStar goal model being constructed in the tool.

Fig. 5 presents the functionality that provides a workflow for adding iStar model elements. Users can select from predefined templates for key elements like actors, intentions, and dependencies. The tool then guides users through inputting the required information for each selected element type. This allows modelers to quickly add multiple elements in a structured way before moving on to graphical modeling and refinement. These adding functions can accelerate iStar practice and meet practical needs by enabling modelers to efficiently input multiple elements and relations at once, providing structured guidance that covers a wide range of iStar modeling scenarios, and facilitating the transition from documentation to modeling via automated extraction and addition.

Moreover, we design the auto-layout functionality to provide a clearer view of modeling and automate the visualization after batch additions. This can reduce modeling costs and improve efficiency.

7. Conclusions

In this paper, we use empirical research and grounded theory to address the challenges of learning and practicing the iStar modeling framework. Our empirical study utilizes revised Bloom's taxonomy to design research questions and involves ten iStar learners and three iStar lecturers. The analysis of the study results using the coding paradigm of the grounded theory leads to a systematic theory of these difficulties, consisting of detailed strategies as requirements to facilitate the learning and practicing of iStar. Moreover, we implement a prototype tool inspired by this study to bridge the gap between theoretical understanding and practical application further.

Moving forward, we plan to conduct extensive empirical evaluations of the prototype tool in various educational settings to assess its effectiveness in improving iStar learning outcomes. We also aim to refine our theoretical framework by incorporating feedback from a larger and more diverse group of learners and educators, ultimately contributing to the broader field of modeling education.

CRediT authorship contribution statement

Tong Li: Writing – review & editing, Supervision, Project administration, Methodology, Investigation, Funding acquisition, Conceptualization. **Qixiang Zhou:** Writing – original draft, Validation, Software, Investigation, Conceptualization. **Yunduo Wang:** Writing – original draft, Methodology. **Haonan Xiong:** Writing – review & editing, Software. **Wenxing Liu:** Writing – review & editing, Validation. **Ning Ge:** Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Tong Li reports financial support was provided by the National Natural Science Foundation of China. Tong Li reports financial support was provided by the Project of Beijing Municipal Education Commission. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Funding: This work is partially supported by the Central Guidance for Local Scientific and Technological Development Fund (No. 2024ZY0124) and the National Natural Science Foundation of China (No. 62162051).

[1] http://60.205.179.21/prototype/
[2] https://zenodo.org/records/10888884

Data availability

We have shared the data via external links.

References

[1] F. Dalpiaz, X. Franch, J. Horkoff, Istar 2.0 language guide, 2016, arXiv:1605.07767.
[2] 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, Requir. Eng. 24 (2) (2019) 133–160.
[3] E.S.-K. Yu, Modelling Strategic Relationships for Process Reengineering (Ph.D. thesis), University of Toronto, 1995.
[4] T. Li, A.M. Grubb, J. Horkoff, Understanding challenges and tradeoffs in istar tool development, in: Proceedings of the Ninth International I* Workshop (IStar 2016), CEUR Workshop Proceedings, vol. 1674, 2016, pp. 49–54.
[5] A. Mavin, P. Wilkinson, S. Teufl, H. Femmer, J. Eckhardt, J. Mund, Does goal-oriented requirements engineering achieve its goal? in: 2017 IEEE 25th International Requirements Engineering Conference, RE, IEEE, 2017, pp. 174–183.
[6] J. Pimentel, E. Santos, B. Santos, C. Borba, J. Paes, C. Lima, A. Bezerra, J. Castro, F. Alencar, C. Silva, et al., Using i* and tropos in a software engineering contest: Lessons learnt and some key challenges, in: IStar 2010–Proceedings of the 4 Th International I* Workshop, 2010, p. 81.
[7] I. Hadar, I. Reinhartz-Berger, T. Kuflik, A. Perini, F. Ricca, A. Susi, Comparing the comprehensibility of requirements models expressed in Use Case and Tropos: Results from a family of experiments, Inf. Softw. Technol. 55 (10) (2013) 1823–1843.
[8] I.A. ElSayed, Z. Ezz, E. Nasr, Goal modeling techniques in requirements engineering: A systematic literature review, J. Comput. Sci. 13 (9) (2017) 430–439.
[9] Y. Wang, J. Du, T. Li, A learner-friendly approach for using the istar modeling framework: an ongoing study, in: Proceedings of the 14th International IStar Workshop (IStar 2021), CEUR Workshop Proceedings, vol. 2983, 2021, pp. 42–48.
[10] E. Gonçalves, I. Monteiro, Reporting the usage of istar in a model-based industrial project to evolve an e-commerce application, in: Proceedings of the 14th International IStar Workshop (IStar 2021), CEUR Workshop Proceedings, vol. 2983, 2021, pp. 8–14.
[11] 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, vol. 1954, 2017, pp. 48–57.
[12] F. Dalpiaz, J. Horkoff, J. Lockerbie, X. Franch, E. Yu, J. Mylopoulos, et al., Teaching goal modeling in undergraduate education, in: Proceedings of the 1st International IStar Teaching Workshop Co-Located with the 27th International Conference on Advanced Information Systems Engineering, CAiSE 2015, Citeseer, 2015, pp. 1–6.
[13] E. Yu, Towards modelling and reasoning support for early-phase requirements engineering, in: Proceedings of ISRE '97: 3rd IEEE International Symposium on Requirements Engineering, IEEE, Annapolis, MD, USA, 1997, pp. 226–235, http://dx.doi.org/10.1109/ISRE.1997.566873.
[14] L.W. Anderson, D.R. Krathwohl, A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives: Complete Edition, Addison Wesley Longman, Inc., 2001.
[15] B.S. Bloom, M.D. Engelhart, E. Furst, W.H. Hill, D.R. Krathwohl, Handbook I: Cognitive Domain, David McKay, New York, 1956.
[16] D.R. Krathwohl, A revision of Bloom's taxonomy: An overview, Theory Into Pr. 41 (4) (2002) 212–218.
[17] I. Maslov, S. Poelmans, K. Rosenthal, Conceptual modeling education: A bibliometric literature review and avenues for future research, Bus. Inf. Syst. Eng. (2025) 1–33.
[18] T. Nkamaura, Y. Tachikawa, Requirements engineering education using role-play training, in: 2016 IEEE International Conference on Teaching, Assessment, and Learning for Engineering, TALE, IEEE, 2016, pp. 231–238.
[19] C. Gralha, M. Goulao, J. Araujo, Are there gender differences when interacting with social goal models? A quasi-experiment, Empir. Softw. Eng. 25 (2020) 5416–5453.
[20] Y. Zhu, T. Li, An object detection-based framework for automatically recognizing iStar hand drafts, Autom. Softw. Eng. 29 (2) (2022) 57.
[21] H. Singh, H. Khalajzadeh, S. Paktinat, U.M. Graetsch, J. Grundy, Modelling human-centric aspects of end-users with iStar, J. Comput. Lang. 68 (2022) 101091.
[22] M. Ruiz, J. Franch Gutiérrez, Insights from using goal models for teaching data structures, in: Proceedings of the 16th International IStar Workshop (IStar 2023): Co-Located with 31st IEEE International Requirements Engineering 2023 Conference, RE 2023: Hannover, Germany, September 03–04, 2023, CEUR-WS.org, 2023, pp. 47–53.
[23] S.A. Soukour, W. Aboucaya, N. Georgantas, Leveraging knowledge graphs for goal model generation, in: 7th Workshop on Natural Language Processing for Requirements Engineering (NLP4RE) in Conjuction with the 30th International Woking Conference on Requirements Engineering: Foundation for Software Quality, REFSQ 2024, 2024.
[24] B.B. Batista, M.S. Lima, T.U. Conte, Teaching requirements engineering for AI: A goal-oriented approach in software engineering courses, in: Proceedings of the XXIII Brazilian Symposium on Software Quality, 2024, pp. 613–623.
[25] D. Bork, A framework for teaching conceptual modeling and metamodeling based on bloom's revised taxonomy of educational objectives, 2019.
[26] D. Bogdanova, M. Snoeck, Domain modelling in bloom: Deciphering how we teach It, in: The Practice of Enterprise Modeling: 10th IFIP WG 8.1. Working Conference, PoEM 2017, Leuven, Belgium, November 22–24, 2017, Proceedings 10, Springer, 2017, pp. 3–17.
[27] D. Bogdanova, M. Snoeck, CaMeLOT: An educational framework for conceptual data modelling, Inf. Softw. Technol. 110 (2019) 92–107.
[28] C. Wohlin, P. Runeson, M. Höst, M.C. Ohlsson, B. Regnell, A. Wesslén, Experimentation in Software Engineering, Springer Science & Business Media, 2012.
[29] F. Dalpiaz, X. Franch, J. Horkoff, iStar 2.0: A guided tour, 2016, Tutorial slides, URL https://www.dropbox.com/s/4l2k4tbywb8wekk/iStar-tutorial-online.pdf.
[30] 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, ISBN: 9781450375801, 2020.
[31] A. Strauss, J. Corbin, Basics of Qualitative Research, Sage publications, 1990.
[32] D.G. Oliver, J.M. Serovich, T.L. Mason, Constraints and opportunities with interview transcription: Towards reflection in qualitative research, Soc. Forces 84 (2) (2005) 1273–1289.
[33] S. Habchi, X. Blanc, R. Rouvoy, On adopting linters to deal with performance concerns in android apps, in: 2018 33rd IEEE/ACM International Conference on Automated Software Engineering, ASE, 2018, pp. 6–16, http://dx.doi.org/10.1145/3238147.3238197.
[34] B.G. Glaser, A.L. Strauss, E. Strutzel, The discovery of grounded theory; strategies for qualitative research, Nurs. Res. 17 (4) (1968) 364.
[35] K.-J. Stol, P. Ralph, B. Fitzgerald, Grounded theory in software engineering research: a critical review and guidelines, in: Proceedings of the 38th International Conference on Software Engineering, 2016, pp. 120–131.
[36] J. Horkoff, Observational studies of new i* users: Challenges and recommendations, in: iStarT@ CAiSE, 2015, pp. 13–18.
[37] H. Estrada, A.M. Rebollar, O. Pastor, J. Mylopoulos, An empirical evaluation of the i* framework in a model-based software generation environment, in: Advanced Information Systems Engineering: 18th International Conference, CAiSE 2006, Luxembourg, Luxembourg, June 5–9, 2006. Proceedings 18, Springer, 2006, pp. 513–527.
[38] J.P. Carvallo, X. Franch, An empirical study on the use of i* by non-technical stakeholders: the case of strategic dependency diagrams, Requir. Eng. 24 (2019) 27–53.
[39] i* Wiki, i* Tools, URL http://istarwiki.org/tiki-index.php?page=i%2A+Tools.
[40] J. Horkoff, Y. Yu, S. Eric, OpenOME: An open-source goal and agent-oriented model drawing and analysis tool, iStar 766 (2011) 154–156.
[41] J. Horkoff, N.A. Maiden, Creative leaf: A creative istar modeling tool, in: iStar, 2016, pp. 25–30.
[42] J. Lockerbie, N. Maiden, REDEPEND: Extending i* modelling into requirements processes, in: 14th IEEE International Requirements Engineering Conference, RE'06, IEEE, 2006, pp. 361–362.
[43] G. Grau, X. Franch, S. Ávila, J-prim: A java tool for a process reengineering i* methodology, in: 14th IEEE International Requirements Engineering Conference, RE'06, IEEE, 2006, pp. 359–360.
[44] J. Pimentel, J. Castro, Pistar tool–a pluggable online tool for goal modeling, in: 2018 IEEE 26th International Requirements Engineering Conference, RE, IEEE, 2018, pp. 498–499.
[45] A. Edwards-Jones, Qualitative Data Analysis with NVIVO, Taylor & Francis, 2014.
[46] J. Corbin, A. Strauss, Basics of Qualitative Research, third ed., SAGE, Los Angeles, 2008.